On Delivery Guarantees and Worst-Case Forwarding Bounds of Elementary Face Routing Components in Ad Hoc and Sensor Networks

Hannes Frey and Ivan Stojmenovic, *Fellow*, *IEEE*

Abstract—In this paper, we provide a thorough theoretical study on delivery guarantees, loop-free operation, and worst-case behavior of face and combined greedy-face routing. We show that under specific planar topology control schemes, recovery from a greedy routing failure is always possible without changing between any adjacent faces. Guaranteed delivery then follows from guaranteed recovery while traversing the very first face. In arbitrary planar graphs, however, a proper face selection mechanism is of importance since recovery from a greedy routing failure may require visiting a sequence of faces before greedy routing can be restarted again. We provide complete and formal proofs that several proposed face routing and combined greedy-face routing schemes guarantee message delivery in specific planar graph classes or even in arbitrary planar graphs. We also discuss the reasons why other methods fail to deliver a message or even end up in a loop. In addition, we investigate the behavior of face routing in arbitrary not necessarily planar networks and show, while delivery guarantees cannot be supported in such a general case, most face and combined greedy-face routing variants support at least loop-free operation. For those variants, we derive worst-case upper bounds on the number of forwarding steps.

Index Terms—Ad hoc network, sensor network, face routing, correctness analysis, worst-case analysis.

◆

1 INTRODUCTION

Wireless *multihop ad hoc networks* [1] are defined by wireless network nodes communicating without using a fixed network infrastructure. Due to limited communication ranges, sending a message from source to destination often requires collaborating intermediate forwarding nodes. Limited battery capacity and limited overall communication bandwidth mandate that message forwarding which is also referred as *routing* has to be performed in a resource-efficient manner.

1.1 Localized Geographic Routing

Geographic routing [2], [3], [4] forms a specific class of routing protocols, which requires that each network node is able to determine its coordinates by means of a *location system* like GPS or relative positioning based on signal strength estimation [5]. Each routing step requires knowledge about the location of the message's final destination. When the destination location is not known in advance, it has to be requested by using a *location service* [6], which provides a mapping from node addresses to their physical locations.

The majority of geographic routing protocols enable message forwarding in a *localized* manner, i.e., deciding the next routing hop is based solely on a constant amount of information stored in the message, the location of the current node, it's neighbors, and the message's final destination.

A desirable feature of any routing protocol is that it is *loop free* and that it provides *delivery guarantees*. The first one means that under no circumstances, messages may start circulating in the network. The latter refers to the ability of successfully forwarding a message from source to destination. This requires, however, that source and destination are connected by at least one path in the network and an idealized MAC layer, where messages are not lost during any forwarding step.

Elementary geographic routing algorithms employ the *greedy routing* principle by sending the message to the neighbor node, which locally looks best regarding the destination position and the metric being optimized [7], [2], [8]. The most frequently applied metric is to select the node that minimizes the euclidean distance to the destination.

For each localized greedy routing variant, the message may end up in a node that has to drop the message in order to prevent a routing loop. Dropping a message might even be necessary although there exists a path from source to destination node. For this reason, greedy routing is often applied in combination with a *recovery strategy* that is responsible for handling the message as long as greedy routing fails.

1.2 Elementary Face Routing Components

Face routing [9], [10], [11], [12] is the most prominent recovery strategy preserving the stateless property of geographic greedy routing mechanisms. The basic idea is to forward a message along one or possibly a sequence of

---

- *H. Frey is with the Computer Networks Group, Institute of Computer Science, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany. E-mail: hannes.frey@upb.de.*
- *I. Stojmenovic is with SITE, University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada. E-mail: ivan@gmail.com.*

Manuscript received 10 Sept. 2008; revised 14 Apr. 2009; accepted 18 Oct. 2009; published online 12 May 2010.
Recommended for acceptance by S. Fahmy.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2008-09-0458.
Digital Object Identifier no. 10.1109/TC.2010.107.

0018-9340/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

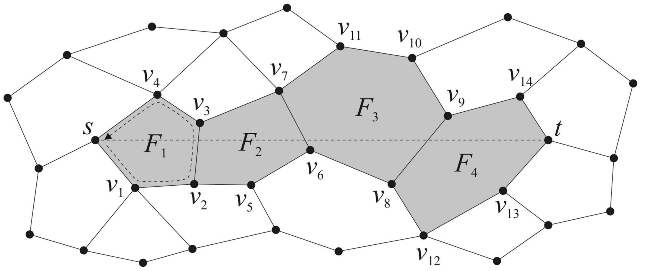


Fig. 1. A message visits a sequence of faces providing progress toward the destination. Each face is handled according to the left or right-hand rule.

adjacent faces of a planar graph such that the visited faces are providing progress toward the destination node. In this connection, a two-dimensional geometric graph is termed *planar* if any two edges intersect in their endpoints only (see Fig. 1 for an example).

The edges of a planar graph constitute polygons, which partition the plane into several inner and one outer face. The basic idea of planar graph routing is to forward a message along a sequence of adjacent faces, which are providing progress toward the destination node t, e.g., the sequence F_1, F_2, F_3, F_4 depicted in Fig. 1. Exploration of a single face can be done in a localized way by applying the well-known *left-hand rule* that requires the message to traverse the edge which is lying next in counterclockwise direction from the previous visited edge. The *right-hand rule*, in contrast, requires the message to travel the edge lying next in clockwise direction. For an example of message forwarding according to the left and right-hand rules, refer to face F_1 in Fig. 1. Applying the right-hand rule and starting with the edge sv_1 will result in the cycle $sv_1v_2v_3v_4s$. Applying the left-hand rule and starting with the edge sv_4 will result in the reverse cycle $sv_4v_3v_2v_1s$.

1.3 Messageless Planar Graph Construction

In general, an arbitrary wireless network graph is not planar. Thus, before planar graph routing can take place, a planar graph construction mechanism (also referred as planar topology control mechanism) has to be applied in advance. Techniques to construct a planar graph locally can be classified into *messageless* ones [13], [14], [15] and *nonmessageless* ones [16], [17], [18], [19], [20]. We term those techniques as messageless which can be applied by a node without further message exchange as soon as this node knows its own, and the location of all its one-hop neighbors. In nonmessageless strategies, in contrast, after getting the one-hop neighbor information, nodes have to exchange further messages in order to agree on a consistent view on the planar topology.

1.4 Contradicting Statements on Delivery Guarantees

Guaranteed delivery of face and several combined greedy-face routing schemes under certain planar graph types is assumed as an established fact. Numerous experimental studies confirm it, and some high-level formal arguments were presented. However, a recent study [21] claimed that the routing protocols GPSR [10], GFG [9], and GOAFR+ [12] cannot guarantee delivery in arbitrary undirected planar network graphs. On the other hand, the formal arguments given for GPSR and GFG did not exploit any structural properties of the planar graph they have been used on. Thus, according to the arguments given in these publications, GFG and GPSR should work in any planar graph. Who is right now?

1.5 Contribution

In this work, we consider all existing face routing variants. We apply them on the known message-free planar graphs and arbitrary network graphs and study their correctness and worst-case behavior with respect to the three most elementary building blocks, which any of these variants has to implement:

- The choice of the first edge and one of both hand rules when face traversal starts.
- The decision when to change from the current traversed face into the next one.
- The choice of the first edge and one of both hand rules when the next face has been selected.

In the subsequent section, we will first summarize how the elementary building blocks are realized in localized protocols described so far. In addition, we will summarize the messageless planar graph constructions known so far. This short survey establishes the base for elaborating the following list of selected contributions.

First, we confirm the findings of [21] and show in addition that the GPSR face routing mechanism may also fail in any of the messageless planar graph constructions. Moreover, we show that the greedy-face routing technique used by GPVFR may fail in arbitrary planar graphs. Those schemes were initially expected to work correctly under any circumstance, which motivates a technical treatment of delivery guarantees of face and combined greedy-face routing schemes.

Second, for face routing under arbitrary planar graphs, we specify two generic building blocks for selecting the first edge and traversal rule when face routing is to be started and when the next face is to be selected. Practitioners should use this specification as a provable correct ground truth when implementing any of the existing or new face routing variants.

Third, we show that all known messageless planar graph constructions support a simplified way of greedy failure recovery: just follow the very first face. This means that when realizing face recovery in this case, there is no need to struggle with any localized face changing rules but one can just implement the simplified face recovery algorithm provided in this work.

Fourth, we give the first complete proofs showing which of the existing face and combined greedy-face mechanisms work correctly under which of the considered planar topologies or under arbitrary planar graphs. For instance, we show that GFG works under any planar graph while GPSR does not. While intuitively face routing looks correct, complete proofs investigating the subtle details were missing so far. Our intention here is just to establish a mathematically correct and complete reference of correctness.

Fifth, we show that even in nonplanar graphs, a message will ever return to the face exploration start node. This insight is of high practical relevance. It says that even if in arbitrary graphs, it is not possible to provide delivery guarantees with a purely localized algorithm [22], face routing can be employed in arbitrary nonplanar graphs as a localized loop-free forwarding heuristic, though.

Sixth, for all face routing variants, we derive worst-case upper bounds on the number of routing steps. For some, we can even show tightness of these bounds. We derive these bounds both for planar graphs and arbitrary not necessarily planar ones. Such bounds are useful to estimate the worst-case behavior in terms of hop count and routing delay. Another practical issue is to use these bounds for dimensioning the messages' maximum time to live values.

Seventh, we describe sufficient conditions for satisfying the upper bounds on routing steps. This means, by showing that the sufficient conditions are met, the upper bound results of this work also apply to the new developed techniques.

2 EXISTING TECHNIQUES SUMMARIZED

2.1 Face Routing Variants

As a general classification, we can distinguish between face routing strategies, which require that the message has to follow a sequence of adjacent faces that are intersected by the straight line st connecting the source node s with the destination node t. These strategies will be denoted by *continuative strategies* since they keep the line st as a reference during the whole routing process. In contrast, a *volatile strategy* will initialize planar graph routing each time a face change occurs. In other words, the node where a face change has occurred is treated as a planar graph routing start node again. According to this definition, the first three of the following listed strategies are continuative while the remaining three are volatile ones.

Greedy-face-greedy (GFG) [9]. As soon as a message encounters an edge which intersects the source destination line st at an intersection point p, it will change into the face which intersects with the open line segment pt. However, only those intersection points are considered that are closer to the destination than the last encountered intersection point where current face traversal was started.

Compass routing II (CR) [11]. A possible alternative to the previously described strategy is to explore the complete face and to advance the message to the edge which intersects st at the point being closest to the destination compared to all intersections encountered during traversal of the current face.

Greedy perimeter stateless routing (GPSR) [10]. A simplified variant of the GFG protocol strictly employs the left-hand traversal rule (the same definition is possible for the right-hand rule as well). When face exploration encounters the next closer intersection, the first edge of the next visited face is determined by simply choosing the edge lying in counterclockwise direction from the intersected edge.

Other face routing (OFR) [12]. A possible variant of following faces that are intersected by the source destination line st is to completely explore the current face and to restart face traversal at the visited node which is located closest to the destination t.

Other face routing (OFR^*) *[23].* A slight modification is to find the closest point on the face boundary instead of the closest node. The message is sent to one of the two endpoints of the edge, which contains the face boundary's closest point. Face traversal is started at this node by treating this node as the face routing start node.

Greedy path vector face routing (GPVFR) [24]. After starting in a node s, a face change occurs as soon as an intersection with st is found. However, this method does not keep the source destination line st but restarts face exploration at the node that encountered the intersection and treats this node as the new start node of the next source destination line st.

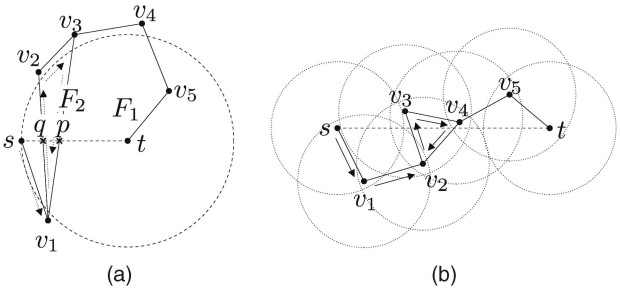


Fig. 2. Examples of GPSR routing failures. (a) The routing failure of GPSR observed in [21]. (b) The face routing variant of GPSR applied on its own has no guaranteed delivery in RNG, GG, and PDT.

2.2 Message-Free Localized Planar Graph Construction

All known localized planar topology control mechanisms require the underlying network to be a *unit disk graph* or *quasi unit disk graph*. A unit disk graph connects any two nodes if and only if their euclidean distance is less than or equal to a unique sending radius r. In other words, the transmission area of each device v is a circle with center v and a unique radius r. The class of quasi unit disk graphs is a generalization of unit disk graphs, which allows a limited variation of the transmission area of each device; it may be of any shape provided that its boundary lies within a minimum and maximum circle while the ratio over the minimum and maximum circle radii is limited by $\sqrt{2}$ [18], [19].

Relative neighborhood graph (RNG) [13]. A node u preserves all outgoing edges uv, which satisfy that the intersection of the circles with center u, center v, and radii $|uv|$ contains no other node than v.

Gabriel graph (GG) [14]. A node u preserves all outgoing edges uv, which satisfy that the smallest circumcircle $C(u,v)$ around u and v—also termed Gabriel circle in the following—contains no other neighbor node than v.

Partial Delaunay triangulation (PDT) [15]. Each node u checks for any other one-hop neighbor v if the Gabriel circle $C(u,v)$ contains any other neighbor node. If not, the edge is a Gabriel graph edge and is kept. Otherwise, two cases are considered. If $C(u,v)$ contains at least one neighbor left and one right of the straight line uv, the edge uv is not kept. If the nodes are only located on one side, the edge is kept if the following condition holds: Let w be the node in $C(u,v)$, which maximizes the size of the circumcircle $C(u,v,w)$. The edge is kept if and only if node u is able to reach all nodes in $C(u,v,w)$ and no neighbor node of u is lying within that circle.

3 ROUTING FAILURES

3.1 Routing Failures of GPSR

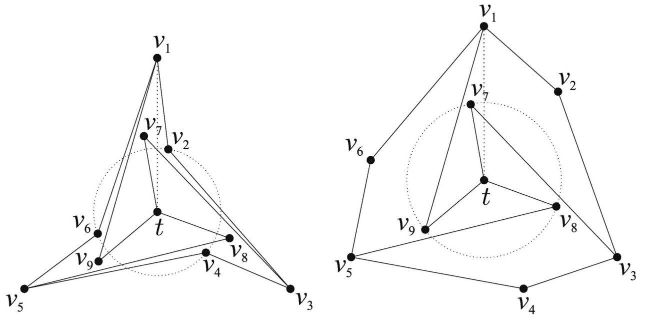
The face routing component of GPSR is an implementation of the before crossing variant, which strictly applies the left or the right-hand rule during the whole face recovery process. Recent studies under arbitrary planar networks show that this variant may produce forwarding errors. For completeness, in Fig. 2, we repeat the GPSR routing failure reported in [21]. Suppose that a message is to be sent from node s to t. Node s will immediately begin with face recovery since its single neighbor node v_1 is not closer to t. When starting with the left-hand rule—i.e., when selecting the next edge lying in counterclockwise direction from the source destination line—the message will traverse the outer face F_1 along the edge sv_1 and encounter the next edge v_1v_3 which is intersecting with st in p. GPSR changes to the inner face F_2 by selecting the next edge in counterclockwise direction and maintaining the left-hand traversal rule. The next traversed edge v_1v_2 is intersecting st as well. However, an additional face change will not occur since the distance between t and this intersection point q is greater than the distance between t and the intersection point p, where the last face change occurred. After changing into the inner face F_2, the message will follow the cycle $v_1v_2v_3v_1$ without finding any intersection point that is located closer to t. Since none of the three nodes v_1, v_2, and v_3 is located closer to t than the start node s, the message will loop without returning into greedy mode again, and finally, be dropped when the first edge v_1v_2 of the current face traversal is traversed again.

As depicted in Fig. 2b, when applying the face routing component of GPSR without greedy routing, a routing failure may as well occur in any of the discussed localized planar graph constructions. In fact, none of the planar graph construction methods RNG, GG, and PDT will remove an edge from the unit disk graph depicted in Fig. 2b. Similar to the error discussed for Fig. 2a, running the GPSR face routing mechanism in this example will end up in the cycle $v_2v_3v_4v_2$.

What is the reason behind those routing failures? If the face routing part of GPSR would have been applied on source destination pair (s,t) in Fig. 1, the destination would have successfully be reached along the path $sv_4v_3v_7v_{11}v_{10}v_9v_{14}t$. When comparing that successful routing example with the routing errors presented in Fig. 2, the following can be observed. In Fig. 1, the path segments of a message handled according to the left-hand rule are always lying above the straight line st. On the other hand, when applying the left-hand rule in the start node s in Fig. 2, the message accidentally moves behind the line segment st and follows the path which is located below the straight line st.

3.2 Routing Failures of GPVFR

It turns out that the GPVFR face routing variant may end up in a routing loop when applied in an arbitrary planar graph. For example, suppose that in Fig. 3a node v_1 addresses a message to t. Face traversal may either start with edge v_1v_2 or edge v_1v_9. The variant described in [24] will select the edge v_1v_2 since node v_2 is closer to t than v_9. This leads to the path $v_1v_2v_3$ until encountering the edge v_3v_7, which is intersected by the source destination line v_1t. Thus, according to the GPVFR variant, planar graph routing simply restarts in node v_3. Similarly, the message will be forwarded along the path $v_3v_4v_5$ until encountering the edge v_5v_8, which is intersected by the current source destination line v_3t. Restarting face routing at v_5 finally results in the path $v_5v_6v_1$ while v_1 restarts face routing again. Thus, we have a routing loop although there exists a path from source to destination. It is obvious that selecting from both edges the one instead whose endpoint is farther from the destination (i.e., edge v_1v_9 instead of edge v_1v_2 in Fig. 3a) will not solve the problem in general. An appropriate example where this variant will fail is depicted in Fig. 3b. Since node v_2 is farther away from t than it is v_9, face exploration will again start with edge v_1v_2, which leads to the same loop as it was depicted in Fig. 3a.



(a) (b)

Fig. 3. Planar graph routing according to the GPVFR strategy might end up in a routing loop in arbitrary planar graphs. (a) Loop example for the variant selecting the node closer to t. (b) Loop example for the variant selecting the node far away from t.

The graph in Fig. 3a is as well an example that even the combination of greedy routing and GPVFR-based planar graph routing from [24] might end up in a routing loop. Suppose that node v_2 initiates routing of a message toward the destination node t and that the node v_3 is marginally closer to t than v_1. Since node v_2 has no neighbor closer to t, it will immediately start face exploration and will select the edge endpoint v_3 from both possible edges v_2v_1 and v_2v_3 as the first routing step. The node v_3 will detect the intersection between v_3v_7 and the source destination line v_2t and will restart face routing with the source destination line v_3t. From thereon, the same routing loop occurs as described before. This follows from the fact that the message never visits a node which is located closer to t than v_2 where recovery was started. Thus, the message will never return into greedy mode again.

4 BUILDING CORRECT FACE ROUTING VARIANTS

In the following, we describe two algorithms, which should be used as a basis block to build continuative and volatile face routing variants which provably work. We will start to describe an algorithm which should be applied by continuative strategies whenever a face change has to be performed due to an encountered intersection with the source destination line. After that, we describe an algorithm which should be applied by a continuative or a volatile strategies in order to select the face on face traversal start.

4.1 A General Face Selection Rule

Whenever face routing decides to select the next face, the node currently forwarding the message may select one of the two possible starting edges in order to explore the new face. In general, a continuative variant switching to next face traversal at node v, and forwarding the message along the edge which is not intersected by st, is denoted by *before crossing variant*. In contrast, the *after crossing variant* will select the remaining other edge. A third alternative, the *best angle variant* always selects the edge vw, which minimizes the angle $\angle wvt$, notwithstanding if vw is intersecting with the source destination line or not. In other words, from the current node v's local view, the best angle method selects

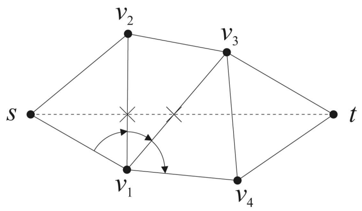


Fig. 4. Before crossing variant.

the next hop node, which minimizes the path deviation from the line vt.

Refer to Fig. 1 for an example of the described variants. When starting with the edge sv_1, the complete routing path from start node s to the destination node t corresponds to $sv_1v_2v_5v_6v_8v_{12}v_{13}t$ according to the before crossing variant, $sv_1v_2v_3v_7v_6v_8v_9v_{14}t$ according to the after crossing variant, and $sv_1v_2v_5v_6v_8v_9v_{14}t$ according to the best angle variant.

The observation we discussed for GPSR in Section 3.1 suggests that when exploration of a face is started, one has to check which one of the two half planes defined by the source destination line st will be traversed at the beginning, and then choose between the left or right-hand rules accordingly. In the following, we will show that it is sufficient to check whether the destination node t is located left or right of the encountered edge intersecting the source destination line. More precisely, for a given node v and an outgoing edge vw, we denote t to be located on the left-hand side of vw if the angle between vw and vt in counterclockwise direction is smaller than 180 degree. Otherwise, if the angle between vw and vt in clockwise direction is smaller than 180 degree, we say that t is located on the right-hand side. Note that the case vw being collinear with vt is not covered by this definition.

For the after crossing variant, the generic face selection rule basically works as follows: Suppose that face routing decides to traverse the next face due to the next visited edge vw intersecting the source destination line st. When sending the message along the intersected edge vw, traversal of the next visited face has to be performed according to the left-hand rule if the destination node t is located on the right-hand side of vw. Otherwise, if it is located on the left-hand side, exploration has to be performed according to the right-hand rule. In a similar way, we can determine the correct edge and rule in order to forward the message according to the before crossing variant. Here, we can use the fact that the face currently traversed can as well be traversed in the opposite direction by switching between the left and right-hand rules. In other words, when the after crossing variant selects the intersected edge vw and the left-hand rule, we obtain the opposite face traversal direction by using the right-hand rule and starting with the edge lying in clockwise direction from vw. In the same way, when the after crossing variant selects the right-hand rule, opposite face traversal is obtained by applying the left-hand rule and starting with the edge lying next in counterclockwise direction from vw.

As depicted in Fig. 4, the before crossing variant requires some additional attention since the selected next edge may intersect st as well. For instance, starting in node s and applying the right-hand rule leads to message forwarding from s to v_1. The next visited edge v_1v_2 will intersect the straight line st connecting the source and destination nodes. The next edge v_1v_3 lying in clockwise direction from v_1v_2 and following the right-hand rule, however, intersects with the straight line st as well. By selecting the edge v_1v_4 lying in clockwise direction from the currently investigated edge v_1v_3, we finally have found an edge which does not intersect st. In general, when starting from the encountered intersecting edge, we have to investigate a clockwise or counterclockwise sequence of intersected edges until finding the first one that is not intersected by st.

Summarizing the discussion, the following pseudocode (see Algorithm 1) provides the essential building block in order to determine the traversal rule and the next hop node. The algorithm requires that the message arrives at a node u not located on st while the next explored edge uv is *regularly* intersected by the source destination line st. Two edges intersect regularly if they have only one intersection point in common. After applying the code for this case, the variables `rule_before` and `rule_after` store the traversal rule (i.e., left or right-hand rule) required for the before and after crossing variants, respectively. The variables `node_before` and `node_after` store the next hop node, which is responsible to forward the message accordingly.

Algorithm 1. Calculating the next node and traversal direction in case of a regular intersection of uv with the source destination line

Require: current node u, intersecting edge uv, and destination node t

Ensure: `node_before`, `rule_before`, and `node_after`, `rule_after` store the node and rule for before and after crossing face traversal, respectively

```
if $ut$ is located right of $uv$ then
    set $rule$ to right hand rule
else
    set $rule$ to left hand rule
end if
set $w$ to $v$
repeat
    set $v$ to $w$
    let $uw$ be the edge next to $uv$ according to $rule$
until $uw$ does not intersect $st$
set rule_before to $rule$
set node_before to $w$
set rule_after to the opposite of $rule$
set node_after to $v$
```

4.2 Face Exploration Start

We still have to consider which traversal rule and outgoing edge has to be selected when face exploration is started for the first time. In addition, during face exploration, we might encounter a node which is located on the source destination line st. Then, Algorithm 1 will not be able to find an outgoing edge not intersecting with st. The special case that the current node is located on st can be handled by simply restarting face routing from such a node. Thus, in the following, it is sufficient that we elaborate only one algorithm which supports both a start node and such a special node. A possible solution is given by Algorithm 2. The result variables `node_rhr` and `node_lhr` store the next

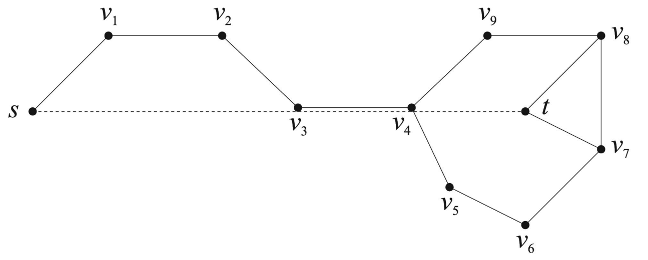


Fig. 5. The endpoint of an edge lying on the line st may be handled as a face routing start node.

hop nodes for exploring the face according to the right or left-hand rule, respectively.

Algorithm 2. Starting or restarting face routing in case of a node which is located on the source destination line

```
Require: current node $u$, and destination node $t$
Ensure: node_rhr and node_lhr store the node for
    right-hand and left-hand rule traversal, respectively
let $uv$ be the edge minimizing $\angle vut$
if $ut$ is located right of $uv$ then
    let $uw$ be the next edge in clockwise direction
    set node_rhr to $w$
    set node_lhr to $v$
else
    let $uw$ be the next edge in counter clockwise
        direction
    set node_rhr to $v$
    set node_lhr to $w$
end if
```

Refer to Fig. 5 for an example of the described procedure. Suppose that the message is currently stored in node v_4. The outgoing edge v_4v_9 minimizes the angle regarding the line v_4t. Since v_4t is located right of v_4v_9, we apply the left-hand rule when forwarding the message along v_4v_9. In order to forward the message in the opposite direction, we choose the opposite rule, i.e., the right-hand rule, and select the edge v_4v_5 which is located next to v_4v_9 according to the right-hand rule.

In addition, a message visiting a node v located on the source destination line st may encounter an edge vw being collinear to the line st. In this case, vw and st have an infinite number of intersection points in common, which we denote as an *irregular intersection*. An example of this kind of intersection is given by edge v_3v_4 and st depicted in Fig. 5. Irregular intersections can simply be handled by ignoring all intersection points except for the edge endpoint. For instance, according to this definition, the outgoing edge v_3v_4 of node v_3 intersects the line st at the point v_4. When face traversal is restarted at a node v having an outgoing edge vw which is collinear to the source destination line st, the line vt is neither located on the left or the right-hand side of vw. When starting with the edge vw, we can select between the two faces having the boundary edge vw in common. Thus, the next selected rule may be one of both possible ones.

5 DELIVERY GUARANTEES

From a top level point of view, proving guaranteed delivery of face and combined greedy-face routing seems rather be an obvious task. However, as we pointed out in the previous sections, when it comes to the specific details, proving delivery guarantees deserves a closer look. The purpose of this section is to provide a technical sound and complete investigation including those details and using the definition of planarity and the definition of faces as the only basic requirement.

We start our investigations by having a closer look on the structure of unit disk graphs and the discussed localized planar topologies. This forms the foundation for the subsequent observations on some of the combined greedy-face routing algorithms. Subsequently, we show correctness of the elementary building blocks described by Algorithms 1 and 2. The results will, in turn, be used in order to show which face routing variants provide guaranteed delivery under which graph assumptions. Finally, we conclude this section by discussing the properties of combined greedy-face routing and proving respective results on delivery guarantees of such combined methods.

5.1 Structural Graph Properties

It can be observed that RNG, GG, and PDT support delivery guarantees of some combined greedy-face variants, which do not have such guarantees for arbitrary planar graphs. This observation is due to the following property:

Definition 1. *A graph G satisfies the* progress property *if the following is satisfied. Let st be the line between any two graph nodes s and t in S. For any edge $uv \in G$ intersecting the line st, the distance between t and at least one of the edge endpoints u or v is smaller than the distance between s and t.*

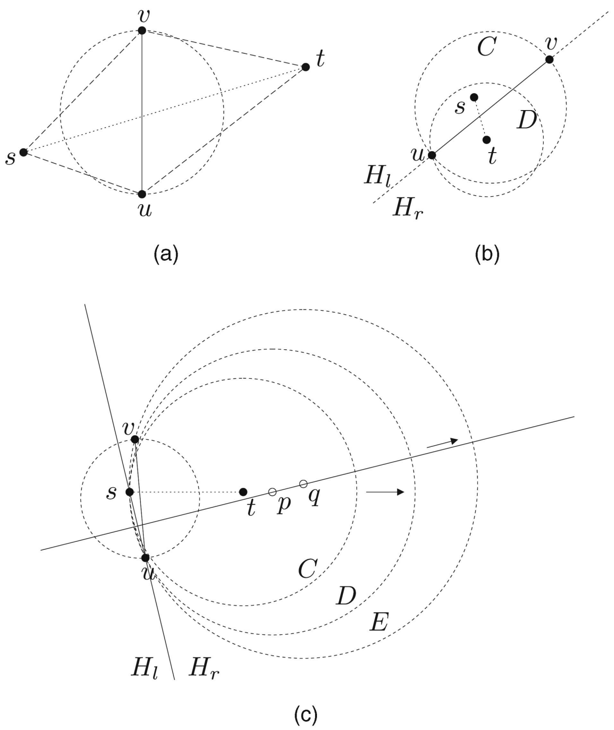
Theorem 1. *The partial Delaunay triangulation PDT satisfies the progress property.*

Proof. Let S be a finite node set, st be the line between any two network nodes s and t in S, and uv be any edge in $PDT(S)$ which intersects the line st. We consider three cases to show that the distance between t and at least one of the edge endpoints u or v is smaller than the distance between s and t.

Case 1: The Gabriel circle $C(u,v)$ does not contain s or t (see Fig. 6a). It follows that both angles $\angle usv$ and $\angle utv$ are less than 90 degree. Since the angles of the quadrilateral (s,u,t,v) sum up to 360 degree, at least one of the angles $\angle sut$ and $\angle svt$ has to be greater than 90 degree. It follows that at least one of the two nodes u or v is located closer to t than the node s.

Case 2: The Gabriel circle $C(u,v)$ contains t (see Fig. 6b). The straight line uv partitions the plane into two half planes H_l and H_r. Since uv and st are intersecting, w.l.o.g., we can assume that $s \in H_l$ and $t \in H_r$. Assume for the sake of contradiction that $|st| \leq \min\{|ut|,|vt|\}$ is satisfied. Without loss of generality, we assume that $|ut| \leq |vt|$. Let D be the circle with center t and radius $|ut|$. We have that s is located in D. It follows that $s \in D \cap H_l \subseteq C \cap H_l$. Thus, the circumcircle $C = C(u,v)$ contains nodes on both sides $C \cap H_l$ and $C \cap H_r$. If follows that uv is not an edge of $PDT(S)$, which is a contradiction.

Case 3: The Gabriel circle $C(u,v)$ contains s (see Fig. 6c). Assume for the sake of contradiction that $|st| \leq \min\{|ut|,|vt|\}$ is satisfied. Let C be the circle with center t and radius $|st|$. Without loss of generality, we assume that $|ut| \leq |vt|$. Let H_l and H_r be the two half planes defined by the straight line su. Without loss of generality,



(a) (b) (c)

Fig. 6. Face exploration of a PDT topology always finds a node closer to the destination. (a) Nodes s and t are not in $C(u, v)$. (b) Node t is located in $C(u, v)$. (c) Node s is located in $C(u, v)$.

we assume that t is located in H_r. It follows that v is located in H_r as well. Otherwise, st and uv would not intersect.

Consider now the circle D with s and u on its boundary and with center p on the straight line st. Due to $|ut| \geq |st|$, we have $C \subseteq D$. Moreover, due to $|ut| \leq |vt|$, we have v which is located outside or on the boundary of D. It follows that $|up| \leq |vp|$.

Consider now the circumcircle E around u, v, and s. Due to $|up| \leq |vp|$, we have $D \cap H_r \subseteq E \cap H_r$. It follows that $t \in C \cap H_r \subseteq D \cap H_r \subseteq E \cap H_r \subseteq E$.

Due to $s \in C(u, v)$, node s is visible to u and v. Since uv is an edge of $PDT(S)$, the circumcircle E must satisfy $E \subseteq C(u)$, with $C(u)$ being the unit disk around u. This implies, however, that node t is visible to node u. It follows that the edge uv will not be included in $PDT(S)$, which is a contradiction. □

Corollary 1. *Relative neighborhood graph RNG and Gabirel graph GG satisfy the progress property.*

Proof. This follows immediately from Theorem 1 and the subset relation $RNG \subseteq GG \subseteq PDT$, i.e., each edge in those subgraphs is an edge in PDT as well. □

Unit disk graphs are mandatory for most of the planar topology control schemes. Thus, it is interesting to investigate the structural property of the unit disk graphs as well. In fact, independent of the planar graph construction method, we can observe the following easy-to-prove property:

Lemma 1. *Let s and t be two nodes which are not connected in a unit disk graph G and let uv be an edge in G which intersects with st. If u and v satisfy $|vt| < |st| < |ut|$, then node s is always connected to v.*

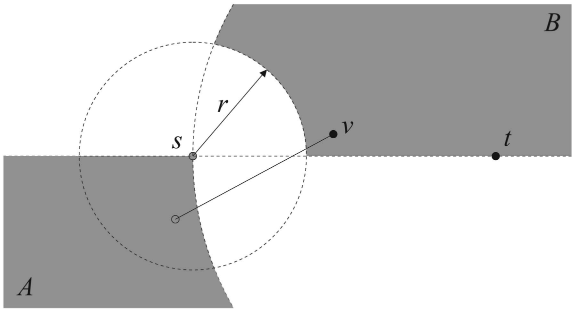


Fig. 7. The edge endpoint v is always connected to s.

Proof. Since uv is intersecting st, we can assume, without loss of generality, that u is located below and v is located above the straight line st (see Fig. 7). We assume that node v is not connected to node s. Due to $|vt| < |st|$, node v has to be located in the gray-shaded area B. Since $|ut| > |st|$, node u is located in the gray-shaded area A. Since s and t are not connected, each line l connecting the areas A and B satisfies $|l| > r$ while r is the unit disk radius. It follows that u and v are not connected, which is a contradiction. □

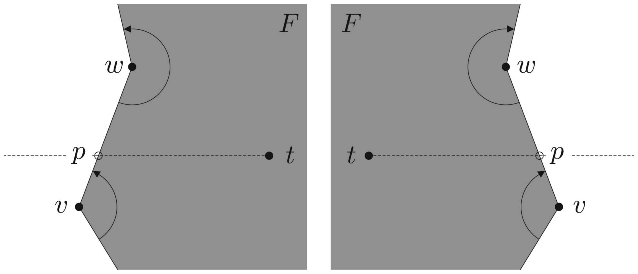
5.2 Correctness of the Building Blocks

We motivated by appropriate examples that the described methods for starting face exploration and switching between two adjacent faces are reasonable building blocks in order to provide delivery guarantees. However, we still have to prove that this is indeed the general case. In the following, we use the term "interior" of a face in order to refer to all points located within a face excluding the face boundary. An immediate consequence of this definition is that a node v located on the face boundary cannot be connected to a node w located in the face interior. Otherwise, w would be located both within the face interior and the face boundary. The following easy-to-prove Lemma 2 describes a generalization of this observation. This forms the basis to prove Lemma 3 and Lemma 4, which finally provides the foundation in order to prove guaranteed delivery of face routing when applying the described face starting or face selection rule described by Algorithms 2 and 1, respectively:

Lemma 2. *Let F be the face of a planar graph G and t be a node located in the interior of F. There exists no path in G which connects t with a node on the face boundary.*

Proof. We assume for the sake of contradiction that there exist a node s on the face boundary and a path $w_1 \ldots w_m$ leading from node $s = w_1$ to node $t = w_m$. Since the path endpoint t is located in the face interior, we can find an index k with w_k not located and w_{k+1} located in the interior of F. By definition, w_{k+1} is not located on the face boundary. In addition, w_k is not located on the face boundary since no node from the face boundary is connected to a node in the face interior. Thus, $w_k w_{k+1}$ is intersecting the face boundary with w_k and w_{k+1} being different from the nodes of the face boundary. It follows that there exists one face boundary edge vw, which intersects with $w_k w_{k+1}$ in a point different from their endpoints. This contradicts the planarity of G. □

Lemma 3. *Let s and t be a pair of nodes, which are reachable in a planar graph G. Let vw be an edge reachable from s and*



(a) (b)

Fig. 8. When the reachable destination t is located on the left/right-hand side, applying the right-/left-hand rule will always find an intersection point which is closer to t than the current intersection point p. (a) On destination, on the right-hand side, the left-hand rule has to be applied. (b) On destination, on the left-hand side, the right-hand rule has to be applied.

intersecting st *in a point* $p \neq t$. *When starting in* v, *face exploration according to the rule and first edge determined by Algorithm 1 will always encounter an edge which intersects* st *in a point* q *with* $|qt| < |pt|$.

Proof. Let F be the face traversed according to the left/right-hand rule when starting with edge vw. By assumption, we have that s and v are reachable in G. In addition, reachability of s and t implies reachability of v and t as well. It follows by Lemma 2 that t is not located inside the face F. Consider the case that vt is located on the right-hand side of vw and that the face is followed according to the left-hand rule (see Fig. 8a). When the face boundary does not intersect the line st in a point q with $|qt| < |pt|$, the line segment pt excluding the point p is located completely in F. For the opposite case vt being located on the left-hand side and applying the right-hand rule (see Fig. 8b), the same assumption implies that pt excluding p is located in F. In both cases, we have that the node t is located in F, which is a contradiction. □

Lemma 4. *Let* s *and* t *be a pair of nodes, which can reach each other in a planar graph* G. *When starting in node* s, *face exploration according to the start edge and rule determined by Algorithm 2 will always encounter an edge which intersects* st *in a point* $p \neq s$.

Proof. Let F be the face traversed according to the outgoing edge and rule determined by Algorithm 2. Since the selected outgoing edges are the closest ones in clockwise and counterclockwise directions, the intersection between st and F is more than $\{s\}$. Suppose that st does not intersect the face boundary at any other point $p \neq s$. It follows that t is located in the interior of F. By Lemma 2, s and t cannot reach each other, which is a contradiction. □

Corollary 2. *Let* s *and* t *be a pair of nodes, which can reach each other in a planar graph* G *which satisfies the RNG, GG, or PDT property. When starting in node* s, *face exploration according to the start edge and rule determined by Algorithm 2 will always encounter a node* v *which satisfies* $|vt| < |st|$.

Proof. By the previous Lemma 4, we have that face exploration starting in node s will always encounter an edge e intersecting st in a point $p \neq s$. By Theorem 1 and Corollary 1, it follows that one of the edge endpoints v satisfies $|vt| < |st|$. □

5.3 Face Routing Strategies

In the following, we will first have a closer look on delivery guarantees of face routing algorithms when applied on their own. These results will subsequently be used in order to show that as well the combination of greedy and these face routing variants provide delivery guarantees:

Theorem 2. *The face routing variants employed by GFG and CR guarantee delivery in any planar graph.*

Proof. The proof for GFG and CR is the same if we generally consider that the next face is selected at any edge that encountered an intersection located closer to t than the previous one; this may be the first one (GFG) or the one forming the closest intersection (CR). If we determine exploration of the next face at either a node located on or a node not located on the source destination line st, by applying the appropriate algorithm 2 or 1 and due to Lemmas 4 and 3, we always encounter an edge that intersects st in a point which is located closer to t than the previous intersection where exploration of this face was stared at. Since we have a finite number of network edges, face exploration will eventually reach an edge e, which intersects st at point t. By the planarity of G and since t is a node in G, t has to be one of the endpoints of e. □

Theorem 3. *The face routing variant* OFR* *guarantees delivery in any planar graph.*

Proof. A proof on guaranteed delivery can be found in [23, Lemma 4.1]. The authors prove the claim from a top level point of view simply stating that the next face is chosen such that it always contains points that are nearer to t. The details of locally determining the first edge and the correct traversal rule are not covered. Algorithm 2 and Lemma 4 of this work are forming a supplement making the proof in [23] complete. □

Theorem 4. *The face routing variant OFR guarantees delivery in RNG, GG, and PDT.*

Proof. By Corollary 2, face exploration according to the start edge and traversal direction selected by Algorithm 2 will always encounter a node located closer to t than the node where face exploration was started at. Thus, restarting face exploration at the node located closest to t compared to all nodes on the face boundary yields a sequence of nodes $v_1, v_2, \ldots, v_n$, which satisfies $|v_1t| > |v_2t| > \cdots > |v_nt|$. Since the planar graph is finite, this face routing variant will eventually reach the final destination node t. □

Note that if the face routing component of GPVFR is applied in a graph supporting the progress property and if it restarts face exploration using Algorithm 2, by Lemma 4, we have that at least one of both nodes is located closer to t than the node where face exploration was started. However, the other node might, in general, be further away from t than the node where the next face exploration was started. In other words, GPVFR might start at a node which is not located closer to t and we cannot prove delivery guarantees following the idea of the proof for Theorem 4. However, we conjecture that the GPVFR face routing component provides delivery guarantees in graphs, which satisfy the progress property. A proof for this claim is still an open issue.

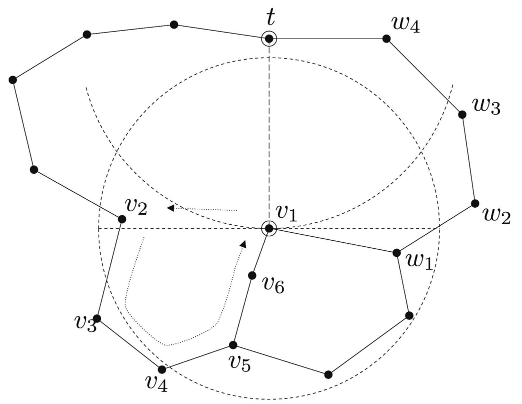


Fig. 9. Combined greedy-face routing may end up in a loop if the employed greedy strategy allows the message to travel in backward direction.

5.4 Combined Greedy-Face Routing

We conclude this section by discussing a class of greedy routing mechanisms, which can be combined with proper face routing mechanisms without loosing delivery guarantees. Moreover, we show that for graphs which satisfy the progress property, delivery guarantees are even possible if defective face recovery mechanisms are employed in combination with these greedy routing mechanisms.

5.4.1 Face Strategies with Delivery Guarantees

In order to preserve their delivery guarantees, proper face routing mechanisms may not be applied in combination with any greedy routing strategy. For instance, this can be observed for two prominent greedy routing mechanisms MFR [7] and GEDIR [25]. The MFR strategy will send the message to the neighbor node whose projection on the straight line st provides maximum progress toward the final destination t. The GEDIR strategy does not consider maximum progress but sends the message to the neighbor node that minimizes the distance toward t. In order to provide loop-free operation, both methods will drop the message if the best choice is the neighbor node which forwarded the message previously.

As depicted in Fig. 9, both strategies cannot simply be combined with face routing without any other additional mechanism in order to prevent a routing loop. Suppose that node v_1 receives or creates a message which is addressed to the destination t. According to MFR or GEDIR, the message will be sent to neighbor node v_2. Both strategies will decide that the previous sender v_1 is the best neighbor of the receiving node v_2 and have to start face routing in order to recover from this greedy routing failure. When employing the right-hand rule, node v_2 will send the message along the path $v_2v_3v_4v_5v_6v_1$ while v_1 is the first node which is located closer to t than the greedy failure node v_2. Thus, greedy routing can be restarted at node v_1, which will, in turn, send the message to v_2 again, i.e., we have constructed a routing loop.

The key observation in this example is that the greedy method MFR or GEDIR allows the message to visit a node whose distance to t is greater than the distance between the previous sender and t. A routing loop in this form is not possible in Fig. 9 if we require that greedy routing has to reduce the distance toward the destination t in each routing step. In this case, node v_1 will start face recovery immediately. By applying the right-hand rule, for instance, the message will be forwarded along the path $v_1w_1w_2w_3w_4$ while w_4 might start greedy forwarding again, finally finding the destination node t. In fact, the observation within this example can be stated as a general sufficient criterion for guaranteed delivery.

Theorem 5. *The combination of greedy routing and any planar graph recovery mechanism provides guaranteed delivery if greedy routing reduces the distance toward the destination node in each routing step and if the applied face recovery mechanism has guaranteed delivery when applied on its own.*

Proof. When starting in a node u, planar graph recovery will either arrive at the destination node t or it will encounter a node v which satisfies $|vt| < |ut|$. Greedy routing started at node v will always encounter a node w, which satisfies $|wt| \leq |vt|$ before face recovery might get started again. Thus, we have an alternating sequence of greedy and face routing executions while each face routing execution is started at a node closer to t than the node where the previous face execution was started. Since we have a finite number of nodes, either the last execution of greedy or face routing will eventually find a path to the destination node. □

5.4.2 Face Strategies without Delivery Guarantees

Although we have shown that the face routing variant GPVFR does not provide guaranteed delivery in arbitrary planar graphs, when used in combination with greedy routing and when applied on a graph which satisfies the progress property, face recovery will run in the first face only and always return into greedy mode before changing into the next face.

Theorem 6. *The entire GPVFR protocol provides guaranteed delivery in RNG, GG, and PDT.*

Proof. Whenever GPVFR uses Algorithm 2 in order to start face routing to recover from a greedy failure node s, Lemma 4 ensures that it will arrive at an edge uv intersecting the source destination line st while at least one of the nodes u or v is closer to t than s. The implementation of the GPVFR protocol returns into greedy mode as soon as a node encounters a neighbor which is located closer to the destination than the node where face routing was started. In addition, the Greedy routing algorithm employed by GPVFR always selects the node that is closest to the destination node t. Thus, the sequence of nodes $v_1, v_2, \ldots, v_n$ recovering from a greedy failure satisfies $|v_1t| > |v_2t| > \cdots |v_nt|$. The finite number of nodes implies that the destination node t will eventually be reached. □

The GPSR implementation returns into greedy mode only if face exploration visits a node which is closer to the destination than the face exploration start node. This different return strategy requires a closer look since Lemma 4 only assures that at least one endpoint of the intersecting edge is located closer to the final destination. The question arises if during recovery GPSR might encounter an intersecting edge with one endpoint which is farther from the destination than the face routing start node, and if it will encounter this edge endpoint first. However, as it is shown in the following proof, this is not possible for topologies which support the progress property and which are build on top of a unit disk graph:

Theorem 7. *The entire GPSR protocol provides guaranteed delivery in RNG, GG, and PDT.*

Proof. Let s be a greedy failure node and t be the destination. According to the proof of Theorem 6, face recovery of GPSR will arrive at a node u having an outgoing edge uv intersecting the source destination line st. GPSR will return into greedy mode if $|ut| < |st|$ is satisfied. Assuming that $|ut| \geq |st|$, by Lemma 4, we have that $|vt| < |st|$ holds. This together with Lemma 1 implies that s and v are connected in the unit disk graph. Finally, $|vt| < |st|$ is a contradiction to s being a greedy failure node. □

5.5 Simplified Face Traversal under the Progress Property

In Algorithm 3, we propose a significant simplification when using face routing as a recovery mechanism for greedy routing and when applying face recovery under planar graphs supporting the progress property. During recovery from a greedy routing failure, the algorithm returns into greedy mode during exploration of the very first face. This is always possible due to the progress property, i.e., this face always contains a node closer to the destination than the node where face exploration was started at. Traversal of the very first face can be done either by the left-hand rule or right-hand rule.

Algorithm 3. Simplified greedy-face routing scheme for planar graphs supporting the progress property.

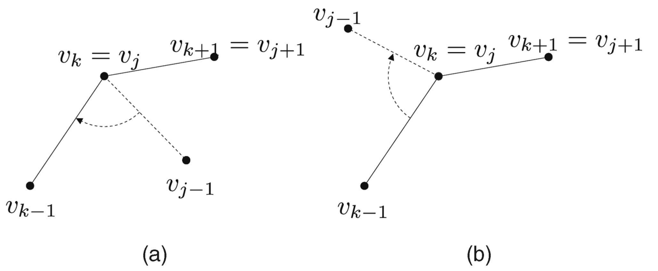
```
repeat
  follow greedy until delivery or failure at node $s$
  if failure at $s$ then
    use Alg. 2 to select face $F$ containing the line $st$
    traverse $F$ until return to greedy is possible or face is visited completely
  end if
until delivery or face visited completely
```

Theorem 8. *Assume that Algorithm 3 is used with a greedy routing strategy, which selects only nodes that are closer to the destination than the current forwarding node. Algorithm 3 provides guaranteed delivery under RNG, GG, and PDT.*

Proof. This follows immediately from Theorem 1, Corollary 1, and Lemma 4. If there exists a path from source to destination, each face traversal ends with visiting a node closer to the destination than the node where face recovery started at. Moreover, greedy routing selects only nodes closer to the destination than the current forwarding node. Thus, the sequence of nodes $v_1, v_2, \ldots, v_n$ recovering from a greedy failure satisfies $|v_1t| > |v_2t| > \ldots |v_nt|$. The finite number of nodes implies that the destination node t will eventually be reached. □

6 WORST-CASE FORWARDING BOUNDS

In this section, we analyze those face routing strategies that support loop-free operation in planar network topologies, i.e., the continuative strategies GFG, GPSR, and CR, and the volatile strategies OFR and OFR*. In arbitrary networks, all of these strategies cannot support guaranteed delivery. This is an immediate consequence of a recent important result, which reveals that finding a deterministic localized unicast forwarding strategy which supports delivery guarantees in arbitrary network graphs is not possible [22]. However, we show in this section that although delivery guarantees are not supported for arbitrary network graphs, the listed face routing components provide at least loop-free operation. In addition, we derive upper bounds on the number of forwarding steps when running those variants in arbitrary graphs. Results for planar graphs follow immediately from those general results.



(a) (b)

Fig. 10. Considering the edges v_jv_{j-1}, v_kv_{k-1}, and v_kv_{k+1} relative to v_jv_{j+1}. (a) Assuming v_jv_{j-1} before v_kv_{k-1} in clockwise direction. (b) Assuming v_jv_{j-1} after v_kv_{k-1} in clockwise direction.

6.1 Face Routing

Theorem 9. *Let G be a finite undirected graph. Assume that starting at an edge e, the graph is explored according to the left-hand or right-hand rule, respectively. The edge e is the first edge which is traversed twice in the same direction.*

Proof. Due to symmetry, we consider the right-hand rule only. Let $\{v_i\}$ be the sequence of visited nodes. Each step of the right-hand rule traversal visits an edge of the graph. Due to the finite number of edges, there will be at least one edge which is eventually traversed twice in the same direction. Let v_kv_{k+1} be the first visited edge in $\{v_i\}$, which is traversed twice in the same direction. It follows that there exists $j < k$ such that $v_jv_{j+1} = v_kv_{k+1}$. Assume that v_jv_{j+1} is not the first edge of the right-hand traversal. The edge $v_{j-1}v_j$ visited before edge v_jv_{j+1} is not the edge $v_{k-1}v_k$ visited before v_kv_{k+1}. Otherwise, this will contradict the assumption that v_kv_{k+1} is the first edge visited twice in the same direction. It follows that $v_{j-1} \neq v_{k-1}$. Assume that relative to v_jv_{j+1}, the edge v_jv_{j-1} is lying in clockwise direction before the edge v_kv_{k-1} (see Fig. 10a). It follows that v_jv_{k-1} is the next edge following the edge $v_{j-1}v_j$ according to the right-hand rule. By assumption, the edge v_jv_{j+1} is the next edge and $v_jv_{j+1} \neq v_jv_{k-1}$, which is a contradiction. Thus, relative to v_jv_{j+1}, the edge v_jv_{j-1} must lie in clockwise direction after the edge v_kv_{k-1} (see Fig. 10b). It follows that $v_jv_{j-1} \neq v_kv_{k+1}$ is the next edge traversed after visiting $v_{k-1}v_k$, which is again a contradiction. It follows that edge v_kv_{k+1} is the first edge of the right-hand rule exploration. □

In other words, the previous Lemma assures that when starting in any edge according to one of both rules, the traversal will never get lost somewhere in the network. It will always return to the edge where traversal started at. Moreover, when considering the edges uv in G as two directed edges (u, v) and (v, u), each such directed edge will be visited at most once and the directed edges visited in one traversal are forming a cycle. Thus, the following definition is well defined:

Definition 2. *Let G be a finite undirected graph. Let $e_1, \dots, e_k$ be directed edges visited by the left/right-hand rule when starting in any of the edges e_i. We define $\{e_1, \dots, e_k\}$ as the* generalized face *according to left-hand or right-hand rule, respectively.*

Theorem 10. *Let $G = (V, E)$ be a finite undirected graph. The routing steps of any face routing strategy are upper bounded by $O(|E|^2)$ if it satisfies the following:*

- *Left-/right-hand rule is not changed while exploring a generalized face.*
- *The message is dropped once the first edge of an explored generalized face is visited twice in the same direction.*
- *At each edge, exploration of a generalized face is started at most once.*

Proof. Since exploration of a generalized face can only be started at most once for each edge, exploration of a generalized face can be started at most $|E|$ times. Since right-/left-hand rule is not changed during exploration of a generalized face and according to Theorem 9, an edge traversed twice in the same direction is always the face exploration start edge uv. It follows that any other edge of G is visited at most twice. In fact, if any other edge is visited three times, it will be traversed twice in one direction, which would contradict uv being the first edge traversed twice in one direction. Thus, exploration of the current generalized face requires at most $2 \cdot |E|$ steps before exploration changes to the next face or before the message is dropped. We get that the total number of routing steps is limited by $2 \cdot |E| \cdot |E|$. □

Corollary 3. *Let $G = (V, E)$ be a finite undirected graph. The routing steps of the GFG or GPSR face routing part are upper bounded by $O(|E|^2)$.*

Proof. This follows immediately from Theorem 10, whose conditions are met by GFG and GPSR. In fact, starting exploration of a new face is only possible when reaching an edge that intersects with the source destination line in a point closer to the destination than all other intersection points encountered so far. It follows that for each graph edge, exploration of a generalized face can only be started at most once. Moreover, by definition, GFG and GPSR are not changing left-/right-hand rule while exploring a generalized face and a message is dropped when face exploration start edge is visited twice in the same direction. □

Theorem 11. *Let $G = (V, E)$ be a finite undirected graph. The routing steps of any face routing strategy are upper bounded by $O(|E|)$ if it satisfies the following:*

- *Left-/right-hand rule is not changed while exploring a generalized face.*
- *A message explores a generalized face at most once and is then either dropped or advanced to another generalized face.*
- *All visited generalized faces are mutually distinct.*

Proof. We show that for any edge $uv \in E$, the directed edge (u, v) is visited at most four times. Assume for the sake of contradiction that (u, v) is visited more than four times. Let F_i be the generalized face, where (u, v) is visited for the first time. With Theorem 9, the edge (u, v) is visited at most two times when handling this face. The first time for traversing the face, and perhaps a second time for advancing to the node where the face change takes place. It follows that there exist at least two later visited generalized faces F_j and F_k, where (u, v) is visited as well.

For any edge $uv \in E$, the directed edge (u, v) is contained in at most two generalized faces, the one we get when traversing along (u, v) according to the left-hand rule, and the one we get according to the right-hand rule. It follows that two of the three visited generalized faces F_i, F_j, and F_k are the same, which is a contradiction. □

Corollary 4. *Let $G = (V, E)$ be a finite undirected graph. The routing steps of CR and OFR are upper bounded by $O(|E|)$.*

Proof. This follows immediately from Theorem 11, whose conditions are met by CR and the OFR. In fact, both will switch from generalized face F_i to F_{i+1} if F_i has a node closer to the destination than the node v_i where exploration of F_i began. The node v_{i+1} where exploration of F_{i+1} starts satisfies $|v_{i+1}t| < |v_i t|$ and $|v_{i+1}t| \leq |vt|$ for all nodes v visited along F_i. Thus, $\min\{|vt| : v \in F_i\} > \min\{|vt| : v \in F_{i+1}\}$. It follows that all visited generalized faces are mutually distinct. Moreover, left/right-hand rules are not changed while exploring a generalized face and messages are either handled in a next face or dropped after full exploration of a generalized face. □

Corollary 5. *Let $G = (V, E)$ be a finite undirected graph. The routing steps of* OFR* *are upper bounded by $O(|E|)$.*

Proof. Similar to the proof of Corollary 4, we only have to show that all generalized faces visited by OFR* are distinct. In fact, OFR* will switch from F_i to F_{i+1} if F_i visits an edge closer to the destination than the edge e_i where exploration of F_i began. The edge e_{i+1} where exploration of F_{i+1} starts satisfies $|e_{i+1}t| < |e_i t|$ and $|e_{i+1}t| \leq |et|$ for all edges e visited along F_i. It follows that all visited generalized faces are mutually distinct. □

Corollary 6. *Let $G = (V, E)$ be a finite undirected planar graph. The number of routing steps of the GFG and GPSR face routing part is upper bounded by $O(|V|^2)$. The number of routing steps of CR, OFR, and* OFR* *is upper bounded by $O(|V|)$.*

Proof. This follows immediately from Corollaries 3, 4, 5, and the fact that the number of edges in a planar graph is upper bounded by $O(|V|)$. □

6.2 Combined Greedy-Face Routing

In this section, we consider the worst-case behavior of combined greedy-face routing. With greedy routing, we refer to any mechanism which reduces the distance to the destination in each routing step.

Theorem 12. *Let $G = (V, E)$ be an arbitrary undirected finite graph. The number of combined greedy-face routing steps is upper bounded by $O(|V||E|^2)$ if face routing satisfies the properties of Theorem 10.*

Proof. The employed greedy routing mechanism reduces the distance between the visited and destination nodes in each forwarding step. Face recovery switches back to greedy only when arriving at a node closer to the destination than the node where greedy recovery started. It follows that the total number of greedy routing steps is upper bounded by $|V|$.

Thus, the number of times face recovery is started is limited by $|V|$. Due to Theorem 10, the total number of forwarding steps is limited by $O(|E|^2)$. Thus, the total number of forwarding steps is upper bounded by $O(|V| + |V||E|^2) = O(|V||E|^2)$. □

Corollary 7. *Let $G = (V, E)$ be an arbitrary undirected finite graph. The number of routing steps of greedy routing combined with any of the face recovery schemes of GFG and GPSR is upper bounded by $O(|V||E|^2)$.*

Proof. The claim follows from Theorem 12, and as shown in the proof of Corollary 3, that the face recovery schemes of GFG and GPSR satisfy the conditions of Theorem 10. □

Theorem 13. *Let $G = (V, E)$ be an arbitrary undirected finite graph. The number of combined greedy-face routing steps is upper bounded by $O(|V||E|)$ if face routing satisfies the properties of Theorem 11.*

Proof. As shown in Theorem 12, the total number of greedy routing steps is upper bounded by $|V|$ and the number of times face recovery is started is limited by $|V|$. Due to Theorem 11, the total number of forwarding steps performed in each face recovery phase is limited by $O(|E|)$. The total number of forwarding steps is thus upper bounded by $O(|V| + |V||E|) = O(|V||E|)$. □

Corollary 8. *Let $G = (V, E)$ be an arbitrary undirected finite graph. The number of routing steps of greedy routing combined with any of the face recovery schemes CR, OFR, or* OFR* *is upper bounded by $O(|V||E|)$.*

Proof. The claim follows from Theorem 13, and as shown in Corollary 4 and Corollary 5, that these recovery schemes satisfy the conditions of Theorem 11. □

Corollary 9. *Let $G = (V, E)$ be an arbitrary undirected finite planar graph. The number of routing steps of greedy routing combined with any of the face recovery schemes used in GFG or GPSR is upper bounded by $O(|V|^3)$. The number of routing steps of greedy routing combined with any of the face recovery schemes CR, OFR, or* OFR* *is upper bounded by $O(|V|^2)$.*

Proof. This follows immediately from Corollary 7, Corollary 8, and the fact that the number of edges in a planar graph is upper bounded by $O(|V|)$. □

6.3 Tightness of the Bounds

We finally conclude this section by providing examples which show that all upper bounds except for GFG or GPSR in combination with greedy forwarding can be reached. For proving the upper bound for the latter ones, we were using a very rough estimate. We could not find an example graph, where this scheme as well meets the derived upper bound. We conjecture that a tighter upper bound can be derived in this case.

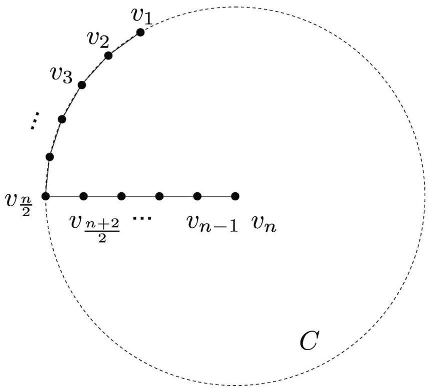


Fig. 11. Example where combined face and greedy routing requires $O(|V||E|)$ forwarding steps.

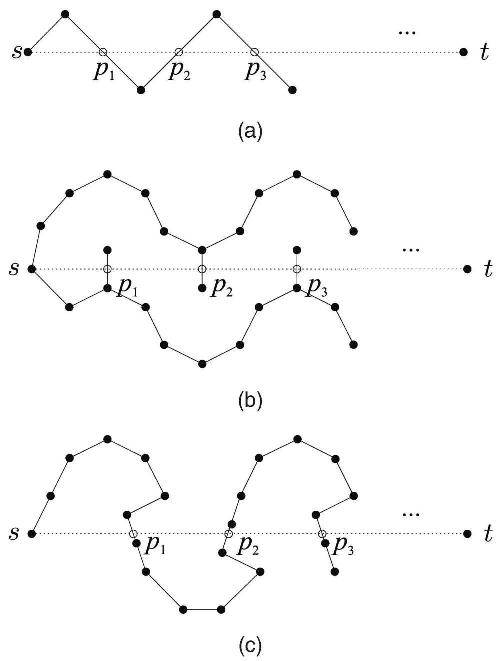
For all the remaining schemes, we construct planar unit disk graphs, which get disconnected if a single edge is removed. Since all discussed planarization methods guarantee connectivity, it follows immediately that such a unit disk graph is invariant to any such localized planarization method. The graphs are constructed in such a way that the upper bound can be reached for at least one source destination pair. This implies that the upper bound can be reached in any of the considered graph classes: arbitrary graphs, planar graphs, and unit disk graphs.

Theorem 14. *For any node number n, there exists a unit disk graph which is invariant to any localized planarization method and where the upper bound $O(|E|)$ of Theorem 11 is reached.*

Proof. For any node number n, consider the graph $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ and $E = \{(v_i, v_{i+1}) : 1 \leq i < n\}$. For any given unit disk graph radius R, we arrange the nodes in a line such that G satisfies the unit disk graph property. The shortest path from v_1 to v_n requires $n - 1$ forwarding steps. Thus, any scheme considered in Theorem 11 requires at least $n - 1$ routing steps to forward a message from v_1 to v_n. The claim follows from $n - 1$ being the number of edges in G. □

Theorem 15. *For any node number n, there exists a unit disk graph with at least n nodes which is invariant to any localized planarization method and where the upper bound $O(|V||E|)$ of Theorem 13 is reached.*

Proof. For any node number n, we construct a graph as depicted in Fig. 11. To ease the following construction, w.l.o.g., we assume that n is divisible by 2, otherwise, perform the following construction for $n + 1$. For any unit disk radius r, we consider a destination node v_n and a circle C with center v_n and radius $R = r\frac{n}{2}$. We place equidistant nodes $v_{\frac{n+2}{2}}, \dots, v_{n-1}$ along a line from v_n to the boundary of C. We place further nodes $v_1, \dots, v_{\frac{n}{2}}$ with equal distance r along a $\frac{1}{2\pi}$ fraction of the boundary of C such that $v_{\frac{n}{2}}$ and $v_{\frac{n+2}{2}}$ are connected. We get the depicted graph.



(a)

(b)

(c)

Fig. 12. Example unit disk graphs where before crossing, after crossing, and best angle variants meet the upper bound. (a) Before crossing variant. (b) After crossing variant. (c) Best angle variant.

We move $v_1, \ldots, v_{\frac{n}{2}}$ by a marginal maximum offset ϵ_1 such that $|v_it| < |v_jt|$ is satisfied for $1 \leq i < j \leq \frac{n}{2}$. After that, we move v_{2i} by a marginal maximum offset ϵ_2 such that $|v_{2i}t| < |v_{2i-1}t|$ and $|v_{2i}t| < |v_{2i+1}t|$. Moreover, we can select the position for v_{2i-1} and v_{2i+1} such that either $\angle v_{2i-1}v_{2i}t < \angle v_{2i+1}v_{2i}t$ or $\angle v_{2i+1}v_{2i}t < \angle v_{2i-1}v_{2i}t$ is satisfied. Thus, for any strategy selecting either the face start edge which maximizes or minimizes the angle toward the destination t, we can always enforce by the right placement of v_{2i} that the edge v_{2i-1} will be selected.

It follows that greedy routing will be restarted at any v_{2i} for $1 \leq i \leq \frac{n}{4}$. At any such v_{2i}, greedy routing fails immediately again and face recovery will be started. This results in the path $v_{2i}v_{2i-1} \ldots v_2v_1v_2 \ldots v_{2i+1}v_{2i+2}$ until the next greedy/face recovery is started at v_{2i+2}. Thus, the number of forwarding steps until $v_{\frac{n}{2}}$ is reached for the first time is $\sum_{i=1}^{\frac{n}{4}} 2i \cdot 2 = O(n^2) = O(|V||E|)$. □

Theorem 16. *For any node number n, there exists a unit disk graph with at least n nodes, which is invariant to any localized planarization method and where the upper bound $O(|E|^2)$ of Theorem 10 is reached.*

Proof. Consider the example graph constructions depicted in Fig. 12. For the before crossing variant, we consider the graph in Fig. 12a. At any new encountered intersection p_i, the face is traversed along the edge which is not intersecting with st. For the after crossing variant, we consider the graph in Fig. 12b. At any new encountered intersection p_i, the face is traversed along the edge which is intersecting with st. For the best angle variant, we consider the graph in Fig. 12c. At any new encountered intersection p_i, the face is traversed along the edge which has the smaller angle with respect to t.

In all three cases, on each new encountered intersection p_i, the whole path back to s and from there back to the next intersection point p_{i+1} is traversed. Thus, for each new encountered intersection point p_j, all the previous intersection points $p_i, 1 \leq i < j$ are visited twice again. Thus, for k intersection points, the number of traversal steps is proportional to k^2. In all examples, the number of edges n is proportional to the number of intersection points k, i.e., $n \leq ck$ for a universal constant c. Thus, the number of traversal steps is $O(|E|^2)$. □

7 SUMMARY

Face routing is a well-known approach to recover from routing failures, which may occur during greedy forwarding. Although the basic idea is easy to describe, face routing has subtle implementation details which have to be considered precisely in order to enable its delivery guarantees. We discussed in detail so far proposed basic face routing components in combination with and without greedy forwarding over message-free planar topologies and arbitrary planar topologies. The discussion is accompanied by proofs, which substantiate delivery guarantees of some of these face routing variants. In addition, we provided appropriate counter examples why other methods will fail under certain circumstances. In addition, we investigated how these routing variants perform in arbitrary two-dimensional networks, which are not necessarily planar. Results for planar graphs follow immediately from these general statements. Our findings show that the considered face routing variants which are loop free in planar graphs are so as well in arbitrary not necessarily planar networks. Finally, for those mechanisms that work loop free, we asked for the worst-case upper bound on the number of forwarding steps. For most of these bounds, we were able to provide examples that these bound can be met.

The bounds derived in this work differ from the already established bounds shown in [23], [12], [19]. Those publications investigated how far GOAFR and GOAFR+ may deviate from the shortest path under planar graphs which comply with the $\Omega(1)$ model or which are limited in degree. Bounds of this type cannot be derived for the other protocols considered in this work, where detours compared to the shortest path may tend to infinity. In addition, for arbitrary planar graphs, and even more, arbitrary not necessarily planar graphs, these bounds do not hold for GOAFR and GOAFR+, either.

Compared to the already established bounds, in this work, we are asking a different question: what is the worst-case forwarding behavior compared to the number of nodes and/or edges of the entire network? Here, answers are possible for all loop-free protocols and under arbitrary planar and even nonplanar graphs.

While we could have focused on planar graphs only, investigating behavior of face routing under arbitrary not necessarily planar graphs has a highly practical relevance. While original hope was that the idea of face routing may some day be extended to arbitrary network graphs or even 3D graphs, theory teaches us that this will never happen [22]. Despite flooding, for any localized routing protocol, a graph can be constructed such that routing will fail. As a consequence, there are situations where just localized rules

TABLE 1
Delivery Guarantees and Worst-Case Upper Bounds of Face Routing Variants

	Delivery Guarantees				**Upper Bounds**
	RNG	GG	PDT	Any	
GFG	ok	ok	ok	ok	$\Theta(\|E\|^2)$
GPSR	drop	drop	drop	drop	$\Theta(\|E\|^2)$
CR	ok	ok	ok	ok	$\Theta(\|E\|)$
OFR	ok	ok	ok	drop	$\Theta(\|E\|)$
OFR*	ok	ok	ok	ok	$\Theta(\|E\|)$
GPVFR	?	?	?	loop	∞

TABLE 2
Delivery Guarantees and Worst-Case Upper Bounds of Combined Greedy-Face Routing Variants

	Delivery Guarantees				**Upper Bounds**
	RNG	GG	PDT	Any	
GFG	ok	ok	ok	ok	$O(\|V\|\|E\|^2)$
GPSR	ok	ok	ok	drop	$O(\|V\|\|E\|^2)$
CR	ok	ok	ok	ok	$\Theta(\|V\|\|E\|)$
OFR	ok	ok	ok	drop	$\Theta(\|V\|\|E\|)$
OFR*	ok	ok	ok	ok	$\Theta(\|V\|\|E\|)$
GPVFR	ok	ok	ok	loop	∞

will never be able to construct a topology that supports delivery guarantees of face routing.

There are two ways out, either as done in [26] to use a nonlocalized approach to construct a topology which supports face routing under any circumstances, or to remain true to the localized approach, as done in this work, and investigate how far one can go with the localized routing. In practical situation, face routing can only work as a heuristic. Good to know then that face routing messages cannot get lost under arbitrary graphs and good to have some idea of the worst-case forwarding behavior.

A summary of our investigations is given in Tables 1 and 2. The tables' first part shows if or if not a scheme supports delivery guarantees under the discussed localized message-free planarization methods or in arbitrary planar graphs; *Any* refers to any planar graph. Here, *ok* denotes guaranteed delivery, *loop* the possibility of a forwarding loop, *drop* the possibility of an incorrect message drop, and *?* that the behavior is not known at the time of writing.

The tables' second part summarizes the worst-case upper bounds on the number of forwarding steps. We mark bounds which can be met by $\Theta(\cdot)$ while we use $O(\cdot)$ for those bounds for which we do not know if they can be met. We use $|V|$ to denote the number of nodes and $|E|$ to denote the number of edges in the network. Whenever the routing variant may end up in a loop, we mark the upper bound with ∞.

ACKNOWLEDGMENTS

This work was supported in part by the NSERC Strategic Grant STPGP 336406-07 and the NSERC Discovery Grant.

REFERENCES

[1] J. Wu and I. Stojmenovic, "Ad Hoc Networks," *Computer,* vol. 37, no. 2, pp. 29-31, Feb. 2004.

[2] G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Technical Report ISI/RR-87-180, Information Sciences Inst. (ISI), 1987.

[3] H. Frey and I. Stojmenovic, "Geographic and Energy Aware Routing in Sensor Networks," *Handbook on Sensor Networks,* I. Stojmenovic, ed., Wiley, 2005.

[4] S. Giordano and I. Stojmenovic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," *Ad Hoc Wireless Networking,* X. Cheng, X. Huang, and D.-Z. Du, eds., pp. 103-136, Kluwer, 2004.

[5] J. Hightower and G. Borriella, "Location Systems for Ubiquitous Computing," *Computer,* vol. 34, no. 8, pp. 57-66, Aug. 2001.

[6] I. Stojmenovic, "Location Updates for Efficient Routing in Ad Hoc Networks," *Handbook of Wireless Networks and Mobile Computing,* I. Stojmenovic, ed., pp. 451-471, Wiley, 2002.

[7] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.,* vol. 32, no. 3, pp. 246-257, Mar. 1984.

[8] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 11, pp. 1122-1133, Nov. 2001.

[9] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third ACM Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL-M '99),* pp. 48-55, Aug. 1999.

[10] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM/IEEE MobiCom,* 2000.

[11] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. 11th Canadian Conf. Computational Geometry (CCCG '99),* pp. 51-54, Aug. 1999.

[12] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad-Hoc Routing: Of Theory and Practice," *Proc. 22nd ACM Int'l Symp. Principles of Distributed Computing (PODC),* pp. 63-72, 2003.

[13] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition,* vol. 12, no. 4, pp. 261-268, 1980.

[14] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology,* vol. 18, pp. 259-278, 1969.

[15] X.-Y. Li, I. Stojmenovic, and Y. Wang, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation," *IEEE Trans. Parallel and Distributed Systems,* vol. 15, no. 4, pp. 350-361, Apr. 2004.

[16] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric Spanner for Routing in Mobile Networks," *Proc. ACM MobiHoc,* pp. 45-55, 2001.

[17] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed Construction of a Planar Spanner and Routing for Ad Hoc Wireless Networks," *Proc. IEEE INFOCOM,* vol. 3, pp. 1268-1277, June 2002.

[18] L. Barriere, P. Fraigniaud, L. Narajanan, and J. Opatrny, "Robust Position-Based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges," *Proc. Fifth ACM Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL-M '01),* pp. 19-27, 2001.

[19] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-Hoc Networks Beyond Unit Disk Graphs," *Proc. ACM DIALM-POMC Joint Workshop Foundations of Mobile Computing,* pp. 69-78, Sept. 2003.

[20] H. Frey, "Geographical Cluster Based Routing with Guaranteed Delivery," *Proc. Second IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '05),* Nov. 2005.

[21] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "On the Pitfalls of Geographic Face Routing," *Proc. Third ACM/SIGMOBILE Int'l Workshop Foundation of Mobile Computing (DIAL-M-POMC),* 2005.

[22] S. Durocher, D. Kirkpatrick, and L. Narayanan, "On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks," *Proc. Ninth Int'l Conf. Distributed Computing and Networking (ICDCN '08),* pp. 546-557, 2008.

[23] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," *Proc. ACM MobiHoc,* 2003.

[24] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information," *Proc. 13th IEEE Int'l Conf. Network Protocols (ICNP '05),* 2005.

[25] I. Stojmenovic and X. Lin, "Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 10, pp. 1023-1032, Oct. 2001.

[26] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," *Proc. USENIX Symp. Network Systems Design and Implementation,* May 2005.

Hannes Frey received the graduate and PhD degrees in mathematical computer science and computer science from the University of Trier, Germany, in 2001 and 2006, respectively. Until April 2008, he worked at the University of Southern Denmark, and until September 2009 at the University of Paderborn, Germany. Since October 2009, he has been affiliated as a junior professor at the University of Paderborn. His research interests are networking algorithms in ad hoc and sensor networks. Within this research field, he is especially interested in researching localized routing and topology control. He has published altogether more than 40 journal articles, conference contributions, and book chapters in the research area of ad hoc and sensor networks. He served in different chairing positions for ADHOC-NOW 2009, IQ2S 2010, and LOCAN and LOCALGOS since 2007. He was a TPC member and external reviewer for more than 30 conferences, including conferences like EWSN, VTC, MASS, CoNext, INFOCOM, and MobiCom. He was a reviewer for more than 20 international journals, for instance, serving regularly as a reviewer for the *IEEE Transactions on Parallel and Distributed Systems* and *Ad Hoc Networks* (Elsevier).

Ivan Stojmenovic received the PhD degree in mathematics. He held regular and visiting positions in Serbia, Japan, United States, Canada, France, Mexico, Spain, United Kingdom (as chair in applied computing at the University of Birmingham), Hong Kong, and Brazil. He is a full professor at the University of Ottawa, Canada. He has published more than 250 different papers, and edited four books on wireless, ad hoc, and sensor networks and applied algorithms with Wiley/IEEE. He is an editor of over a dozen journals, editor-in-chief of the *IEEE Transactions on Parallel and Distributed Systems* (from January 2010), and founder and editor-in-chief of three journals (*Multiple-Valued Logic and Soft Computing*, *Parallel, Emergent and Distributed Systems*, and *Ad Hoc and Sensor Networks*). He has h-index 35 and over 6,000 citations. One of his articles was recognized as the Fast Breaking Paper, for October 2003 (as the only one for all of computer science), by Thomson ISI Essential Science Indicators. He is a recipient of the Royal Society Research Merit Award, United Kingdom. He is the recipient of Excellence in Research Award of the University of Ottawa 2008-2009. He chaired and/or organized over 50 workshops and conferences, and served in over 100 program committees. Among others, he was/is program co/vice chair at the IEEE PIMRC 2008, the IEEE AINA-07, the IEEE MASS-04&07, EUC-05&08, WONS-05, MSN-05&06, ISPA-05&07; founded workshop series at the IEEE MASS, the IEEE ICDCS, and the IEEE DCOSS; and workshop chair at the IEEE MASS-09, ACM MobiCom/MobiHoc-07, and MobiHoc-08. He presented over a dozen tutorials. He is an elected IEEE fellow (Communications Society, class 2008).