# On Delivery Guarantees of Face and Combined Greedy-Face Routing in Ad Hoc and Sensor Networks

Hannes Frey
IMADA, University of Southern Denmark
DK-5230 Odense M, Denmark
frey@imada.sdu.dk

Ivan Stojmenovic
SITE, University of Ottawa
Ottawa, Ontario K1N 6N5, Canada
ivan@site.uottawa.ca

## ABSTRACT

It was recently reported that all known face and combined greedy-face routing variants cannot guarantee message delivery in arbitrary undirected planar graphs. The purpose of this article is to clarify that this is not the truth in general. We show that specifically in relative neighborhood and Gabriel graphs 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 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. A prominent approach is to visit a sequence of faces which are intersected by the line connecting the source and destination node. Whenever encountering an edge which is intersecting with this line, the critical part is to decide if face traversal has to change to the next adjacent one or not. Failures may occur from incorporating face routing procedures that force to change the traversed face at each intersection. Recently observed routing failures which were produced by the GPSR protocol in arbitrary planar graphs result from incorporating such a face routing variant. They cannot be constructed by the well known GFG algorithm which does not force changing the face anytime. Beside methods which visit the faces intersected by the source destination line, we discuss face routing variants which simply restart face routing whenever the next face has to be explored. We give the first complete and formal proofs that several proposed face routing, and combined greedy-face routing schemes do guarantee delivery in specific graph classes or even any arbitrary planar graphs. We also discuss the reasons why other methods may fail to deliver a message or even end up in a loop.

## Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*



## General Terms

Algorithms, Reliability, Theory

## Keywords

Ad Hoc Network, Face Routing, Guaranteed Delivery

## 1. INTRODUCTION

*Wireless multihop ad hoc networks* 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 mandates that message forwarding which is also referred as *routing* has to be performed in a resource efficient manner.

Geographic routing [4, 5, 8] 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 [10]. 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* [21] 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, and the location of the current node, it's neighbors, and the message's final destination. Localized routing protocols can further be classified regarding their *delivery guarantees*. Guaranteed delivery refers to the ability of successfully forwarding a message from source to destination. The definition requires that source and destination are connected by at least one path in the network and that we have 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 [24, 4, 23]. 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. On the other hand, if successful the majority of greedy routing algorithms produce routing paths having a weight that is comparable to the weight of the shortest possible path. For this reason greedy routing is often applied in combination with a *recovery strategy* which is responsible for handling the message as long as greedy routing fails.

*Planar graph routing* which is also referred as *face routing* [2, 11, 14, 15] is the most prominent recovery strategy preserving the stateless property of geographic greedy routing mechanisms. The basic idea is to planarize the network graph in a localized manner and to forward a message along one or possibly a sequence of adjacent faces which are providing progress towards the destination node. This recovery strategy has extensively been studied in *unit disk* and *quasi unit disk* graphs. A unit disk graph connects any two nodes if and only if their Euclidean distance is less 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 extends the concept of unit disk graphs by allowing 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 $\sqrt{2}$ [1, 16].

Guaranteed delivery of face and several combined greedy-face routing schemes in unit disk and quasi unit disk graphs is a well established fact. Numerous experimental studies confirm it, and some formal arguments were presented. However, a recent study [13] claimed that the routing protocols GPSR [11], GOAFR+ [15] and the planar graph routing schemes that they use can not guarantee delivery in arbitrary undirected planar network graphs. Within this article we confirm these findings and explain the reasons behind. We also demonstrate that this is not true in general. Namely, we prove formally that the protocols of the intitial work on face routing, the greedy-face-greedy scheme GFG and its underlying planar routing variant [2], guarantee delivery in arbitrary planar graphs. The correctness of this strategy was already discussed in the original work. The destination node can always be found, since face traversal monotonically minimizes the distance between the source destination line intersections and the final destination. However, the details of locally changing to the right face at an encountered intersection point are not discussed there. Within this work we elaborates the details of locally selecting the right face and provide complete proofs including all these details. Moreover, we also show that in the relative neighborhood and Gabriel graph classes GPSR [11] always returns into greedy mode during exploration of the very first face, while its underlying planar face routing scheme alone does not guarantee delivery because a face change is sometimes erroneously forced.

In the subsequent section we will first present the basic idea of face routing, localized planar graph construction, and the face routing variants, which have been proposed so far. This will be followed by a discussion of the technical details of these face routing variants. Finally, we investigate these methods regarding their delivery guarantees in arbitrary planar graphs and more specific planar graph subclasses. We analyze both the face routing variants applied on their own and applied in combination with greedy routing.

## 2. FACE ROUTING IN A NUTSHELL

The network formed by wireless nodes which are deployed on a plane can be modeled by a two dimensional geometric graph. Each network node $v$ is represented by a point in the plane which follows from the node's location. Two points $v$ and $w$ are connected by a straight line $vw$ if their corresponding network nodes are connected by an edge in the network graph. A two dimensional geometric graph is *planar* if any two edges intersect in their end points only (see Fig. 1 for an example).

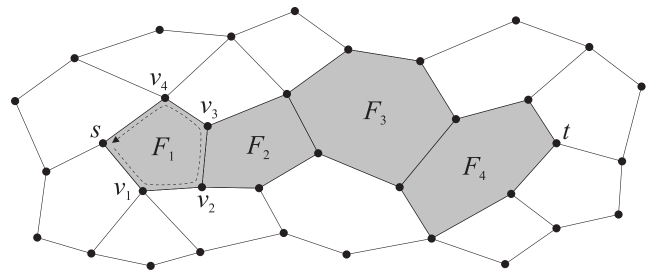


**Figure 1: A message visits a sequence of faces providing progress towards the final destination. Each face is handled according to the left or right hand rule.**

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 the interiors of a sequence of adjacent faces which are providing progress towards 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* or *right hand rule*. Place the left or right hand on one edge of the face boundary and continue exploring the edges by walking along the interior of the face. Message forwarding according to the left hand rule is similar to sending the message along the edge which is lying next in counterclockwise direction from the previous visited edge. The right hand rule in contrast sends the message to the edge lying next in clockwise direction. For an example of message forwarding according to the left and right hand rule 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 $sv_4v_3v_2v_1s$.

When used as a recovery mechanism for a greedy routing failure, planar graph routing may return to greedy routing whenever it encounters a node – this may either be the current message receiver or one of its neighbors – whose distance to the destination is smaller than the distance between the destination and the greedy failure node. The majority of combined greedy and planar graph routing implementations return into greedy mode as soon as possible. However, it was observed that a more elaborate return strategy considering a number of possible nodes before restarting greedy routing can show better performance than returning into greedy mode as soon as possible [15].

Any variant of planar graph routing – applied with or without greedy routing – has to consider the case that there exists no path from source to destination node. In this case the message will eventually traverse the face containing the destination node $t$ without detecting a face providing progress towards $t$. In order to avoid a routing loop in such a case, face routing has to remember the first edge of the current face exploration. The message gets dropped when arriving at the first edge once again without finding a node where greedy routing can be resumed or without finding an adjacent face which provides further progress towards the final destination node.

## 2.1 Constructing a Planar Graph Locally

In general, an arbitrary wireless network graph is not planar. Thus, before planar graph routing can take place, a planar graph construction mechanism has to be applied in advance. In the following we sketch the well known techniques which provide each node a consistent view on the planar graph by removing those outgoing edges which do not match the applied planar graph construction criteria.

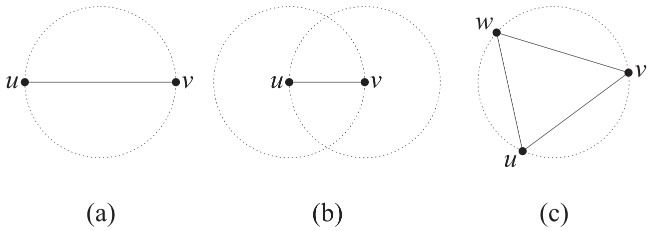


**Figure 2: Localized planar graph construction based on (a) Gabriel graph, (b) relative neighborhood graph, and (c) Delaunay triangulation.**

- *Gabriel graph (GG)* [6] – A node $u$ preserves all outgoing edges $uv$ which satisfy that the circle $U(u, v)$ with diameter $|uv|$ which is passing through $u$ and $v$ contains no other neighbor node than $v$ (see Fig. 2(a)).
- *Relative neighborhood graph (RNG)* [25] – 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 $v$ (see Fig. 2(b)).
- *Localized Delaunay triangulation (LDT)* [7, 19, 20] – Each node computes the Delaunay triangulation on its own neighbor set. The Delaunay triangulation in general contains all triangles which satisfy that the circle passing through the triangle end points does not contain any other node (see Fig. 2(c)). From the subset of outgoing Delaunay edges each node preserves all outgoing edges which are preserved by the node on the other edge end point as well.

Without further modification the listed techniques require the underlying network to be modeled as a unit disk graph. However, by using the concept of virtual edges [1, 16] these methods can be employed in quasi unit disk graphs as well while the above listed structural properties remain the same.

In contrast to GG, RNG, and LDT the recently proposed *cross link detection protocol (CLDP)* [12] can be applied on any network graph. Within this work, however, we do not further investigate this method since it neither operates in a localized way nor always produces a planar graph which is a basic requirement within this work. The fact that CLDP may not produce a planar graph is highlighted in [12, 13] (together with the claim that GPSR can even run without any error on such a resulting non-planar graph). CDLP uses probe messages applying the left/right hand rule until arriving at the start node again. The purpose of these probe messages is to detect any intersections. However, since the number of nodes visited by such a probe message may be arbitrarily large, this protocol is not localized in the sense that we only need information about the nodes in vicinity.

## 2.2 Face Routing Variants

Face routing has been implemented in several variants which differ in the decision when current face traversal has to be interrupted and which face has to be explored next. In the following we list these face routing variants by referring to the well established names of the entire protocols employing these strategies. It is important to note that the following descriptions explain the face routing part of these protocols only. The routing path in the accompanying examples might be different when running the entire protocol. However, as long it is clear from the context we interchangeably use the well established protocol names for both the entire protocol and its face routing part.

As a general classification we can distinguish between face routing strategies which require that the message has to follow a sequence of adjacent faces which are intersected by the straight line $st$ connecting the source node $s$ with the destination node $t$. These strategies will be denoted as *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.

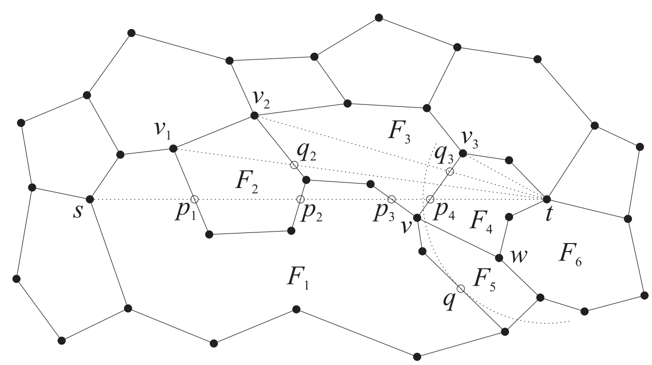


**Figure 3: Continuative and volatile strategies.**

### *Greedy-Face-Greedy (GFG) [2]*

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 which are closer to the destination than the last encountered intersection point where current face traversal was started.

For an example refer to Fig. 3 and suppose that a message is sent from $s$ to $t$. When exploring face $F_1$ according to the left hand rule the message will encounter $p_1$ as the first intersection and will change face exploration from $F_1$ to $F_2$. When exploring the next face $F_2$ the intersection $p_2$ is the only one which is located closer to $t$ than the intersection point $p_1$ where exploration of $F_2$ was started. After the message has arrived at the intersection point $p_2$ it will explore face $F_1$ again. However, the message now only considers intersection points which are located closer to $t$ than the last intersection point $p_2$. Thus, the message will change to face $F_3$ after it encounters the intersection point $p_3$. By exploring $F_3$ the message will arrive at intersection point $p_4$

where the message changes to exploration of face $F_4$, finally reaching the destination node $t$. Altogether, the sequence of visited faces is $F_1, F_2, F_1, F_3, F_4$.

### *Compass Routing II [14]*

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.

For instance, message forwarding from $s$ to $t$ in Fig. 3 will change from face $F_1$ to face $F_3$ directly since the intersection point $p_3$ is the one which is located closest to $t$ compared to $p_1$ and $p_2$. By exploring the next face $F_3$ the message will find $p_4$ as the closest intersection, switch to face $F_4$ and finally reach the destination node $t$ located on this face. Thus, this variant will visit the shorter face sequence $F_1, F_3, F_4$ in this example.

### *Greedy Perimeter Stateless Routing (GPSR) [11]*

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.

Obviously, when strictly applying the left hand rule in Fig. 3, this method will visit the same face sequence as GFG, i.e. $F_1, F_2, F_1, F_3, F_4$. This is due to the fact, that (in the depicted case) on encountering the next edge crossing with $st$ at a point $p$, the next adjacent face which intersects with the open line segment $pt$ can always be traversed by using the left hand rule and selecting from the crossing edge the next one in counterclockwise direction. In other words, the criteria applied in GFG and GPSR are equivalent for this example.

### *Greedy Other Adaptive Face Routing (GOAFR+) [15]*

A possible variant of following faces which 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 $t$.

An example of this strategy can be followed by Fig. 3 again. A message addressed from $s$ to $t$ will find node $v$ as the one located closest to $t$ on the first traversed face $F_1$. Restarting planar graph routing at this node will result in exploration of face $F_4$ where the final destination $t$ can be found. Thus, according to this strategy the message visits the face sequence $F_1, F_4$.

### *Greedy Other Adaptive Face Routing (GOAFR++) [17]*

A slight modification which was implicitly introduced by the authors in their previous work [17] is to find the closest point on the face boundary instead of the closest node. The message is sent to one of the two end points of the edge which contains the face boundary's closest point. Face traversal is started at this node by treating this node as the planar graph routing starting node. (The naming GOAFR++ is not due to the authors original work, however, we use this here in order to distinguish between both possible GOAFR+ variants.)

For instance, in Fig. 3 the message addressed from $s$ to $t$ will find $q$ as the closest point on the boundary of the initially explored face $F_1$. Restarting face exploration at one of end points of the edge containing $q$ will result in exploration of face $F_5$ in both possible cases. The closest point on the boundary of $F_5$ corresponds to the network node $w$. Assuming that node $w$ restarts planar graph routing finally leads to exploration of face $F_6$ where the message is able to find the destination node $t$. Altogether, the message visits the face sequence $F_1, F_5, F_6$.

### *Greedy Path Vector Face Routing (GPVFR) [18]*

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 which encountered the intersection and treats this node as the new start node of the next source destination line $st$.

For an example refer to Fig. 3 and assume that a message from $s$ to $t$ is handled according to the left hand rule. Following this strategy the message encounters the intersection $p_1$ and restarts planar graph routing at node $v_1$ now considering the line $v_1t$ in order to detect the next intersection. By following the left hand rule the message arrives at node $v_2$ detecting the next intersection point $q_2$. Restarting planar graph routing at this node results in exploration of face $F_3$ using $v_2t$ as the source destination line. When applying the left hand rule again, the message will arrive at node $v_3$ where it detects the next intersection point $q_3$. Finally, restarting planar graph routing at this node enables the message to find the destination node $t$ which is located on the last explored face $F_4$. Altogether the message follows the face sequence $F_1, F_2, F_3, F_4$.

## 3. CONTINUATIVE STRATEGIES

Whenever face routing decides to select the next face, the node currently forwarding the message may select one of two possible starting edges in order to explore the new face. For instance, in Fig. 4 a message exploring the face $F_1$ along the path $sv_1v_2$ will arrive at node $v_3$ whose next outgoing edge $v_3v_4$ intersects with the source destination line. If a face change occurs at this node the message may either be forwarded along the edge $v_3v_6$ or the edge $v_3v_4$ in order to explore the next face $F_2$. In general, a continuative variant forwarding the message along the edge which is not intersected by $st$ is denoted as *before crossing variant*. In contrast the *after crossing variant* will select the remaining other edge.

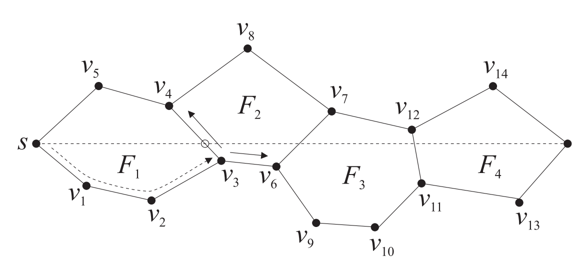


**Figure 4: When changing to the next face there are two possible first edges in order to explore the new face.**

Confer Fig. 4 for an example of the described variants. When starting with the edge $sv_1$ the complete routing path from start node $s$ to destination node $t$ corresponds to

$sv_1v_2v_3v_6v_9v_{10}v_{11}v_{13}t$ according to the before crossing variant, and $sv_1v_2v_3v_4v_8v_7v_6v_9v_{10}v_{11}v_{12}v_{14}t$ according to the after crossing variant.

## 3.1 The Difference between GFG and GPSR

The investigated examples suggest that the after crossing variant requires changing between left and right hand rule each time the message decides to select the next face. On the other hand, it seems that the before crossing variant requires strict application of one of these rules. An implementation of the before crossing variant in this form can be found in the GPSR face recovery part (in the following quoting $D$ refers to the message destination, $x$ refers to the last intersection with the source destination line, and $n$ refers to the chosen next hop): "On each face, the traversal uses the right-hand rule to reach an edge that crosses line $xD$. At that edge, the traversal moves to the adjacent face crossed by $xD$. ... The node forwards the packet along the first edge of this next face – by the right-hand rule, the next edge clockwise about itself from $n$" [11]. In other words, the face is always changed.

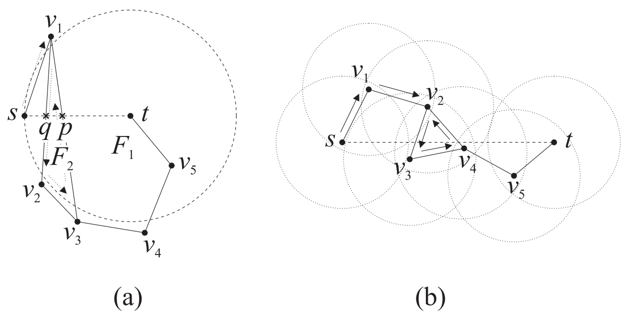


**Figure 5: (a) The routing failure of GPSR observed in [13]. (b) The face routing variant of GPSR applied on its own has no guaranteed delivery in RNG, GG, and LDT.**

Recent studies in arbitrary planar networks show that the face changing rule employed by GPSR may result in a forwarding loop. Suppose that in Fig. 5(a) 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 right hand rule – i.e. when selecting the next edge lying in clockwise 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 clockwise direction and maintaining the right 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 $v_1v_2v_3v_1 \ldots$ without finding any intersection point which 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.

Note that a variant of GPSR which employs the after crossing variant instead will not work in this example as well. Sending the message along the intersecting edge $v_1v_3$ and changing from right to left hand rule results in subsequent traversal of the inner face $F_2$ as well. When traversing the inner face $F_2$ the message will not find a closer intersection point, and will not return into greedy mode again. Thus, the message is caught in the routing loop $v_1v_3v_2v_1 \ldots$.

Note further, that localized Delaunay triangulation allows acute-angled triangles which could be used in order to construct a GPSR routing failure which is similar to the one depicted in Fig. 5(a). However, it is still an open issue if we can maintain the local Delaunay property in each node by connecting the start node $s$ with a single edge $v_1$ similar to Fig. 5(a). Even if this is not possible, there might possibly exist a different local Delaunay triangulation example where the face routing component of GPSR will fail. We conjecture that the face routing component of GPSR has no delivery guarantees in a localized Delaunay triangulation.

As it is depicted in Fig. 5(b), when applying the face routing component of GPSR without greedy routing, a routing loop may as well occur in any of the discussed localized planar graph constructions. In fact, none of the planar graph construction methods, relative neighborhood, Gabriel graph, or localized Delaunay triangulation will remove an edge from the depicted unit disk graph. Similar to the loop discussed for Fig. 5(a), running the GPSR face routing mechanism in this example will end up in the loop $sv_1v_2v_3v_4v_2 \ldots$. As well, the discussed after crossing variant of GPSR will end up in the loop $sv_1v_2v_4v_3v_2 \ldots$.

Due to studies based on the network simulator ns-2 [9] implementation of GPSR it was recently reported that face routing in general can not guarantee delivery in arbitrary planar networks [13]. However, we prove in this article that the delivery of GFG is guaranteed for arbitrary connected planar graphs. This is due to a difference in the face routing procedure between GFG and GPSR. The face routing component of GFG described in [2] only considers intersections with the line $pt$ while $p$ is the last intersection point where a face change has occurred. On face change at the next intersection $p'$ the algorithm proceeds with exploration of the face $F$ that intersects the open line segment $(p', t)$. As it will be proven in Section 5, reachability of $t$ always implies that traversal of the face $F$ will arrive at an edge intersecting the line $p't$ at a point $p''$ with $|p''t| < |p't|$. With this invariant it is easy to show that after a finite number of forwarding steps the message will eventually arrive at the destination node $t$.

When applying GFG in the example depicted in Fig. 5(a) the message will as well be forwarded along the edge $sv_1$ before it detects that the next traversed edge is intersecting with $st$ in point $p$. At this edge, the message continues with exploration of the face which intersects with the remaining line segment $pt$. In other words, in this example the message will continue with exploration of the outer face $F_1$ by sending the message towards node $v_3$ and keeping the right hand rule. Applied with or without greedy routing, the message will eventually reach its final destination by traveling along the path $v_3v_4v_5t$. Similarly, when applying the GFG face routing part in Fig. 5(b), exploration of the same face is kept after the intersection $v_2v_4$ was detected. Thus, GFG is able to deliver the message along the path $sv_1v_2v_4v_5t$.

The subtle component of the GFG top level description is the way a node has to determine, from left or right hand rule, the one which is required in order to traverse the face in the desired direction. A detailed description of the original face routing algorithm which addresses this problem as well can be found in [3]. Later in this article, we present a simplified implementation.

The discussed example shows that there exist cases where the roles regarding maintaining and changing the rotational direction are interchanged for before and after crossing variant. The question arises if it is possible to decide locally when these cases occur. When comparing the successful routing example in Fig. 4 with the routing errors presented in Fig. 5 the following can be observed for both the before and the after crossing variant. In Fig. 4 the path segments where the message is handled according to the left hand rule are always lying above the straight line $st$ while the path segments resulting from the right hand rule are always located below the straight line $st$. On the other hand, when applying the right hand rule in the start node $s$ in Fig. 5 the message accidentally moves behind the line segment $st$ and follows a path which is located above the straight line $st$.

## 3.2 A General Face Selection Rule

The discussed examples suggest when selecting the next face (normally changing, but sometimes the same face is selected) it is not necessary to take the previously applied rule into account. It is rather sufficient to check if the message will traverse the upper or lower half plane defined by the source destination line $st$ and to choose between left or right hand rule 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°. Otherwise, if the angle between $vw$ and $vt$ in clockwise direction is smaller than 180° we say that $t$ is located on the right hand side. Note, 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 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. At this, we can use the fact that the face currently traversed can as well be traversed in the opposite direction by switching between left and right hand rule. 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. 6 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 source and destination node. 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 which is not intersected by $st$.

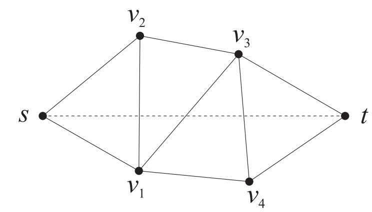


**Figure 6: Before crossing variant.**

Summarizing the discussion, the following pseudo code (see Alg. 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 variant, 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.

**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 $v$

## 3.3 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, Alg. 1 will not be able to find an outgoing edge not intersecting with $st$. It was pointed out in [3] that the special case – 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 Alg. 2. The result variables `node_rhr` and `node_lhr` store the next 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.

```
let uv be the edge minimizing ∠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. 7 for an example of the described procedure. Suppose 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.

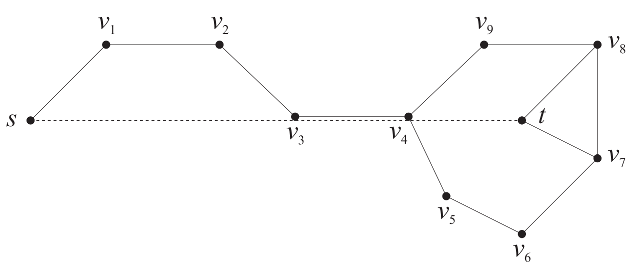


**Figure 7: A node lying on the line $st$ may be handled as a face routing start node.**

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. 7. Irregular intersections can simply be handled by ignoring all intersection points except for the edge end point. 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.

# 4. VOLATILE STRATEGIES

Whenever a volatile face routing variant starts exploration of the next face, the two possible outgoing edges can be determined by employing Alg. 2. Thus, routing errors which were reported for continuative strategies and which are due to a wrong implementation of Alg. 1 are of no concern for these routing strategies. However, from the described volatile strategies only the GOAFR++ variant works well in arbitrary planar graphs.

In the original publication the GOAFR+ variant was studied in conjunction with Gabriel graphs [15] where it has guaranteed delivery. When running the GOAFR+ algorithm in an arbitrary planar graph the message might get dropped although there exists a path from source node $s$ to destination node $t$. For an example refer to the planar graph depicted in Fig. 9(b). When starting in node $s$ the message will explore the interior of the triangle $(s, u, v)$ finding no node located closer to $t$ than $s$. This happens independently from the existence or nonexistence of a path connecting the source node $s$ with destination node $t$. Even when applying the entire GOAFR+ algorithm – its greedy and face routing part – none of the encountered nodes ($u$ and $v$) will be a candidate in order to return to greedy routing, i.e. this routing failure will occur as well. A similar example was reported in [13] where this routing failure was already observed for arbitrary planar graphs. The example in Fig. 9(b) shows furthermore that this variant does not work properly for localized Delaunay triangulations as well.

It turns out that the GPVFR face routing variant may end up in a routing loop when applied in an arbitrary planar graph. For an example suppose that in Fig. 8(a) 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 [18] 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 end point is farther from the destination (i.e. edge $v_1v_9$ instead of edge $v_1v_2$ in Fig. 8(a)) will not solve the problem in general. An appropriate example where this variant will fail is depicted in Fig. 8(b). 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 for Fig. 8(a).

The graph in Fig. 8(a) is as well an example that even the combination of greedy routing and GPVFR-based planar graph routing from [18] might end up in a routing loop. Suppose that node $v_2$ initiates routing of a message towards 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 end point $v_3$ 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 there on the same routing loop occurs as described previously. 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.

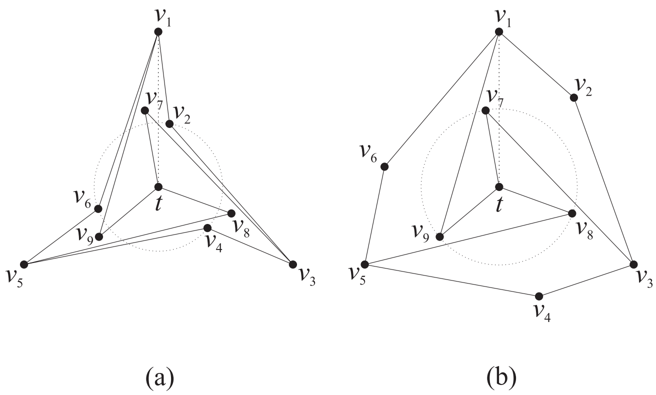


**Figure 8: Planar graph routing according to the GPVFR strategy might end up in a routing loop in arbitrary planar graphs.**

Note, that the GPVFR face routing algorithm might be modified slightly so that it will work for both network graphs depicted in Fig. 8. Instead of restarting planar graph routing at the node which encountered the intersection, the message may choose from both end points of the encountered intersected edge the one which is located closer to the destination $t$ and restart planar graph routing at this node. Referring to Fig. 8, when the message from $v_1$ to $t$ encounters the first intersection between the edge $v_3v_7$ and the source destination line $v_1t$ planar graph routing will be started at node $v_7$ instead of node $v_3$. Depending on the traversal direction the message will reach the destination $t$ either along the path $v_7t$ or the path $v_7v_3v_4v_5v_8t$. If this modified version of GPVFR provides guaranteed delivery for any arbitrary planar graph, however, remains an open issue.

Finally, we point out that the example graphs in Fig. 8 are rather pathological. We will show that such routing failures can not be constructed when using the relative neighborhood or Gabriel graph planarization methods. However, it remains an open issue if a routing failure can be constructed for GPVFR when using localized Delaunay triangulation.

## 5. PROOFS ON 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, relative neighborhood, and Gabriel graphs. 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 Alg. 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 relative neighborhood and Gabriel graphs 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 described property which is satisfied for both graph classes.

Lemma 1. *Let $st$ be the line between any two network nodes $s$ and $t$ in a Gabriel graph $G$. For any edge $uv$ in $G$ intersecting the line $st$, the distance between $t$ and at least one of the edge end points $u$ or $v$ is smaller than the distance between $s$ and $t$.*

Proof. Since $uv$ is a Gabriel graph edge the circle $U(u, v)$ does not contain the network nodes $s$ and $t$ and it follows that both angles $\angle usv$ and $\angle utv$ are less than 90° (see Fig. 9(a)). Since the angles of the quadrilateral $(s, u, t, v)$ sum up to 360°, at least one of the angles $\angle sut$ and $\angle svt$ has to be greater than 90°. It follows, that at least one of the two nodes $u$ or $v$ is located closer to $t$ than the node $s$. □

Corollary 1. *The property of Lemma 1 is as well satisfied for the relative neighbor hood graph.*

Proof. This follows immediately from the fact that RNG is a sub graph of GG , i.e. each edge in a RNG is an edge of the GG as well. □

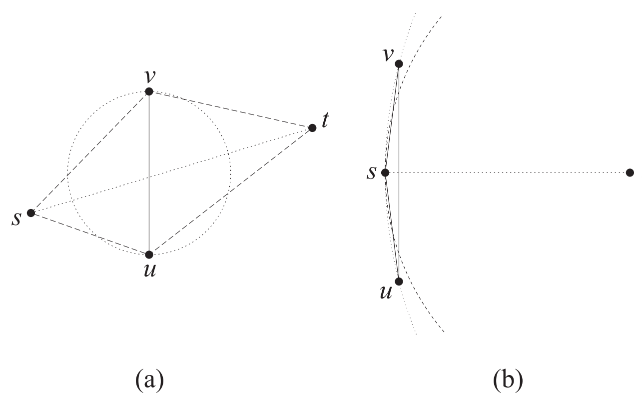


**Figure 9: (a) Face exploration in a Gabriel graph always finds a node closer to the destination. (b) Message forwarding in a LDT face may not find a node closer to the destination.**

The property described in Lemma 1 and its Corollary 1 can not be observed for localized Delaunay triangulation. For instance, in Fig. 9(b) if none of the nodes' neighbors are located in the circle passing the edges of the triangle $(s, u, v)$, the edges $su$, $sv$, and $uv$ will be present in the local Delaunay graph. When considering the source destination line $st$ and the intersecting edge $uv$ we have that it's edge endpoints satisfy $|ut| > |st|$ and $|vt| > |st|$.

Relative neighborhood, Gabriel, and localized Delaunay graph construction requires the underlying network to be modeled as a unit disk graph. This holds as well when applying these graph constructions in quasi unit disk graphs. In this case the quasi unit disk graph is extended by virtual edges so that the resulting supergraph has the unit disk graph property. The planar graph construction methods are then applied on this supergraph. Since unit disk graphs are mandatory for these planar graph classes it is interesting to investigate the structural property of unit disk graphs as well. In fact, independently of the planar graph construction method we can observe the following easy to prove property.

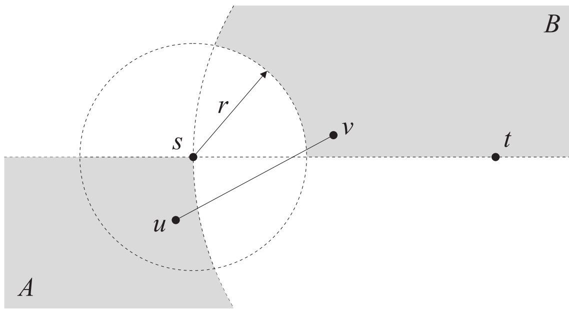


**Figure 10: The edge end point $v$ is always connected to $s$.**

Lemma 2. *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 $v$.*

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. 10). We assume that node $v$ is not connected to node $s$. Due to $|vt| < |st|$ node $v$ has to be located in the grey shaded area $B$. Since $|ut| > |st|$ node $u$ is located in the grey shaded area $A$. Since $s$ and $t$ are not connected, each line $l$ connecting the area $A$ 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 can not 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 3 describes a generalization of this observation. This forms the basis to prove Lemma 4 and Lemma 5 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 Alg. 2 and 1, respectively.

Lemma 3. *Let $F$ be the face of a planar graph $G$ and let $t$ 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 exists 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 end point $t$ 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 end points. This contradicts the planarity of $G$. □

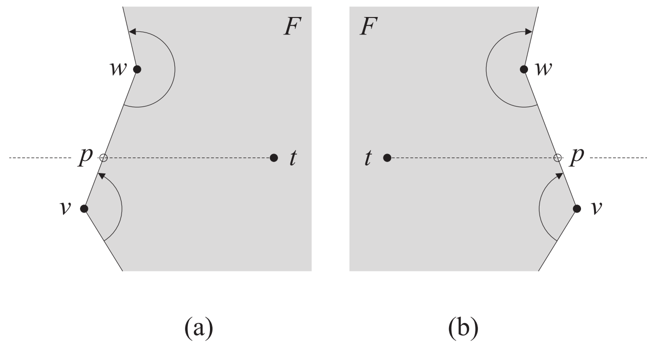


**Figure 11: (a) When the reachable destination $t$ is located on the right hand side applying the left hand rule will always find an intersection point which is closer to $t$ than the current intersection point $p$. (b) The same is satisfied when $t$ is located on the left hand side and when applying the right hand rule.**

Lemma 4. *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 intersecting $st$ in a point $p \neq t$. When starting in $v$, face exploration according to the rule and first edge determined by Alg. 1 will always encounter an edge which is intersecting $st$ 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 3 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. 11(a)). 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. 11(b)) 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 5. *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 Alg. 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 Alg. 2. Since the selected outgoing edges are the closest ones in clockwise and counterclockwise direction, 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 3, $s$ and $t$ can not 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 and let G satisfy the relative neighborhood or Gabriel graph property. When starting in node s, face exploration according to the start edge and rule determined by Alg. 2 will always encounter a node v which satisfies* $|vt| < |st|$.

PROOF. By the previous Lemma 5 we have that face exploration starting in node $s$ will always encounter an edge $e$ intersecting $st$ in a point $p \neq s$. By Lemma 1 and Corollary 1 it follows that one of the edge end points $v$ 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 1. *The face routing variants employed by GFG and Compass Routing II guarantee delivery in any planar graph.*

PROOF. The proof for GFG and Compass Routing II is the same if we generally consider that the next face is selected at any edge which 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 (Compass Routing II). 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 Lemma 5 and 4 we always encounter an edge which 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 end points of $e$. □

THEOREM 2. *The face routing variant employed by GOAFR++ guarantees delivery in any planar graph.*

PROOF. A proof on guaranteed delivery can be found in Lemma 4.1 in [17]. 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 which are nearer to $t$. The details of locally determining the first edge and the correct traversal rule are not covered. Alg. 2 and Lemma 5 of this work are forming a supplement making the proof in [17] complete. □

THEOREM 3. *The face routing variant employed by GOAFR+ guarantees delivery in relative neighborhood and Gabriel graphs.*

PROOF. By Corollary 2 face exploration according to the start edge and traversal direction selected by Alg. 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, if the face routing component of GPVFR restarts face exploration in a relative neighborhood or Gabriel graph using Alg. 2, by Lemma 5 we have that at least one of both nodes are 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 can not prove delivery guarantees following the idea of the proof for Theorem 3. However, we conjecture that the GPVFR face routing component provides delivery guarantees in relative neighborhood and Gabriel graphs. A proof for this claim is still an open issue.

## 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 relative neighborhood and Gabriel graphs delivery guarantees are even possible if defective face recovery mechanisms are employed in combination with these greedy routing mechanisms. On the other hand, in order to preserve their delivery guarantees we show that 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 [24] and GEDIR [22]. The MFR strategy will send the message to the neighbor node whose projection on the straight line $st$ provides maximum progress towards the final destination $t$. The GEDIR strategy does not consider maximum progress but sends the message towards the neighbor node which minimizes the distance towards $t$. 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.

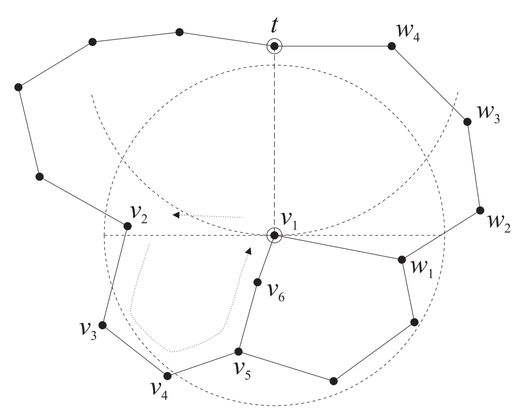


**Figure 12: Combined greedy face routing may end up in a loop if the employed greedy strategy allows the mesage to travel in backward direction.**

As depicted in Fig. 12 both strategies can not 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 addresses to the final 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 methods 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. 12 if we require that greedy routing has to reduce the distance towards the final 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 4. *The combination of greedy routing and any planar graph recovery mechanism provides guaranteed delivery if greedy routing reduces the distance towards 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. □

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 relative neighborhood or Gabriel graph, face recovery will run in the first face only and always return into greedy mode before changing into the next face.

Theorem 5. *The entire protocols GPVFR provides guaranteed delivery in relative neighborhood and Gabriel graphs.*

Proof. Whenever GPVFR uses Alg. 2 in order to start face routing to recover from a greedy failure node $s$, Lemma 5 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 which is closest to the destination node $t$. Thus, the sequence of nodes $v_1, v_2, \ldots, v_n$ recovering from a greedy failure satisfy $|v_1t| > |v_2t| > \ldots |v_nt|$. The finite number of nodes implies that the destination node $t$ will eventually be reached. □

| | RNG | GG | LDT | Any |
|---|---|---|---|---|
| GFG | ok | ok | ok | ok |
| GPSR | loop | loop | loop | loop |
| Compass Routing II | ok | ok | ok | ok |
| GOAFR+ | ok | ok | drop | drop |
| GOAFR++ | ok | ok | ok | ok |
| GPVFR | ? | ? | ? | loop |

**Table 1: Success of face routing applied in its own.**

| | RNG | GG | LDT | Any |
|---|---|---|---|---|
| GFG | ok | ok | ok | ok |
| GPSR | ok | ok | ? | loop |
| Compass Routing II | ok | ok | ok | ok |
| GOAFR+ | ok | ok | drop | drop |
| GOAFR++ | ok | ok | ok | ok |
| GPVFR | ok | ok | ? | loop |

**Table 2: Success of combined greedy-face routing.**

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 5 only assures that at least one end point 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 end point which is farther from the destination than the face routing start node, and if it will encounter this edge end point first. However, as it is shown in the following proof this is not possible for relative neighborhood and Gabriel graphs which require the network to be modeled as a unit disk graph.

Theorem 6. *The entire GPSR protocol provides guaranteed delivery in relative neighborhood and Gabriel graphs.*

Proof. Let $s$ be a greedy failure node and $t$ be the final destination. According to the proof of Theorem 5 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 $|ut| \geq |st|$ by Lemma 5 we have that $|vt| < |st|$ holds. This together with Lemma 2 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. □

# 6. 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 the face routing variants which have been proposed so far. 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. A summary of our investigations is given by Table 1 and 2. At this, *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.

An important observation is that the face routing part of the well established GPSR implementation within ns-2 needs to be revised in order to enable delivery guarantees of this implementation in arbitrary planar networks. We point out as well that this implementation failure is of no concern for simulation runs applying GPSR in combination with relative neighborhood and Gabriel graphs. We have shown that these special cases support guaranteed delivery of GPSR. However, when applying GPSR in planar graphs like considered in [13] correct operation of the face switching strategy becomes essential and should be corrected in the future GPSR implementation.

**Algorithm 3** Simplified greedy-face routing scheme for relative neighborhood and Gabriel graphs.

**repeat**
    follow greedy until delivery or failure at node $s$
    **if** failure at $s$ **then**
        select face $F$ containing the line $st$
        traverse $F$ until return to greedy is possible
    **end if**
**until** delivery

Finally, our analysis shows that under relative neighborhood or Gabriel graphs when recovering from a greedy routing failure it is always possible to return into greedy mode during exploration of the very first face. Provably this face always contains a node closer to the destination than the node where face exploration was started at. Thus, for this specific classes of planar graphs face recovery can be implemented in a simplified way as sketched in algorithm Alg. 3. Traversing that face can be done by left hand, right hand, or alternating left/right hand rule as it is followed by GOAFR+.

# 7. REFERENCES


[1] Lali Barriere, Pierre Fraigniaud, Lata Narajanan, and Jaroslav Opatrny. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *Proceedings of the 5th ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M 01)*, pages 19–27, 2001.

[2] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd ACM International Workshop on discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M 99)*, pages 48–55, Seattle, WA, August 1999.

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

[4] Gregory G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute (ISI), 1987.

[5] Hannes Frey and Ivan Stojmenovic. Geographic and energy aware routing in sensor networks. In Ivan Stojmenovic, editor, *Handobook on Sensor Networks*. Wiley, 2005.

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

[7] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu. Geometric spanner for routing in mobile networks. In *Proceedings of the second ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc '01*, pages 45–55, 2001.

[8] Silvia Giordano and Ivan Stojmenovic. Position based routing algorithms for ad hoc networks: A taxonomy. In Xiuzhen Cheng, Xiao Huang, and Ding-Zhu Du, editors, *Ad Hoc Wireless Networking*, pages 103–136. Kluwer, 2004.

[9] THE CMU MONARCH GROUP. Wireless and mobility extensions to ns-2. http://www.monarch.cs.rice.edu.

[10] Jeffrey Hightower and Gaetano Borriella. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.

[11] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th ACM/IEEE Annual International Conference on Mobile Computing and Networking (MOBICOM-00)*, 2000.

[12] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *Proceedings of USENIX Symposium on Network Systems Design and Implementation*, Boston, Massachusetts, USA, May 2005.

[13] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. On the pitfalls of geographic face routing. In *Third ACM/SIGMOBILE International Workshop on Foundation of Mobile Computing, DIAL-M-POMC*, 2005.

[14] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99)*, pages 51–54, Vancouver, August 1999.

[15] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of the 22nd ACM International Symposium on the Principles of Distributed Computing (PODC)*, pages 63–72, Boston, Massachusetts, USA, 2003.

[16] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Ad-hoc networks beyond unit disk graphs. In *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, pages 69–78, San Diego, September 2003.

[17] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the 4th ACM International Symposium on Mobile Computing and Networking (MobiHoc 2003)*, 2003.

[18] Ben Leong, Sayan Mitra, and Barbara Liskov. Path vector face routing: Geographic routing with local face information. In *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP 2005)*, 2005.

[19] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM '02)*, volume 3, pages 1268–1277. IEEE Computer Society, June 23–27 2002.

[20] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang. Partial delaunay triangulation and degree limited localized bluetooth scatternet formation. *IEEE Transactions on Parallel and Distributed Systems*, 15(4):350–361, 2004.

[21] Ivan Stojmenovic. Location updates for efficient routing in ad hoc networks. In Ivan Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 21, pages 451–471. Wiley, 2002.

[22] Ivan Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, October 2001.

[23] Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, November 2001.

[24] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, March 1984.

[25] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.