

A Survey on Geographic Routing Protocols for Mobile Ad hoc Networks

Atekeh Maghsoudlou, Marc St-Hilaire, and Thomas Kunz

Department of Systems and Computer Engineering

Carleton University, Ottawa, ON, Canada

amagsou@sce.carleton.ca, marc_st_hilaire@carleton.ca and tkunz@sce.carleton.ca

Abstract

Geographic routing has become one of the most suitable routing strategies in wireless mobile ad hoc network mainly due to its scalability. That is because there is no need to maintain explicit routes. The principle approach in geographic routing is greedy forwarding, which fails if the packet encounters a void node (i.e., a node with no neighbour closer to the destination than itself). Face routing and its variations have been proposed and widely studied in the literature as recovery strategies to handle voids. However, face routing strategies are based on two primitives, planarization and face traversal, which make them unsuitable in 3D networks. This survey presents an overview of different face routing algorithms as well as alternatives to face routing strategies. The majority of the proposed face routing strategies and void handling techniques are designed with some idealized assumptions, which are not usually true in realistic scenarios and special types of networks such as sparse networks. We will discuss some of the shortcomings and possible directions for future research from an unmanned aeronautical ad hoc networks' perspective.

Table of Contents

1. Introduction.....	1
2. Geographic Routing.....	4
2.1 Greedy Geographic Forwarding	5
2.2 Recovery Strategies	7
2.2.1 Variations to Face Routing	9
2.2.2 Alternatives to Face Routing	20
3. Discussion	34
4. Randomized Geographic Forwarding (RGF).....	38
4.1 Weight Assignment.....	39
4.2 Discussion	40
5. Conclusion	43
References.....	44

1. Introduction

Ad hoc networks consist of mobile or stationary nodes that communicate over wireless links. There is neither fixed infrastructure to support the communication nor any centralized administration or standard support services. Nodes can self-organize dynamically in an arbitrary and temporary manner allowing people and devices to seamlessly communicate in areas with no pre-existing communication infrastructure; therefore, the nodes themselves act as routers as well. In addition, due to the limited transmission range of wireless nodes, intermediate nodes may be required to collaborate in forwarding a packet from source to destination. Therefore, nodes beyond direct wireless transmission range of each other will be able to communicate via multi-hop routing. The focus in this report is specifically on *Mobile Ad Hoc Networks* (MANETs), where the routing task becomes more challenging due to nodes' mobility. In fact, dynamic topology changes are one of the characteristics of mobile ad hoc networks where nodes move frequently. Therefore, we cannot rely on conventional routing protocols designed for wired networks for the purpose of routing in MANETs. That is why a variety of new routing protocols for mobile ad hoc networks are researched and designed. Some of them, under the category of geographic routing are discussed in this report. There are different scenarios and environments where MANETs are employed; some of them include: military battlefields, disaster recovery and emergency rescue operations, Vehicular Ad Hoc Networks (VANETs) [1] and Aeronautical Networks [2].

One way of communication in such networks might seem to simply flood the entire network. However, the fact that power and bandwidth are scarce resources in such networks of low-powered wireless devices necessitates more efficient routing protocols. Therefore, there are a number of routing protocols proposed for MANETs, which can be categorized into two different approaches: *topology-based* and *position-based* routing [3]. Topology-based routing protocols use information about the links. That is, information about the paths is maintained and routes are established based on the information of the links that exist in the network. These protocols can be further divided into *proactive*, *reactive* and *hybrid* approaches. Proactive protocols are more similar to the classical routing strategies such as distance-vector routing (e.g., Destination-Sequenced Distance-Vector Routing (DSDV) [4]) or link-state routing (e.g., Optimized Link

State Routing (OLSR) [5]). Proactive protocols constantly discover routes and maintain them in routing tables. Hello packets are exchanged periodically by which nodes get informed of changes in the topology. This results in low route discovery latencies at the cost of imposing high overhead due to occupying the bandwidth for route maintenance. On the other hand, reactive protocols discover and maintain routes only if needed, which results in initial delays until the routes are set up. However, the advantage of this type of routing is low overhead in terms of processing and memory along with minimum power consumption and lower bandwidth requirements. In case of topology changes, which result in link failures, route error messages are generated. Although this will be only done for the routes in use, the problem of imposing traffic at times of topology changes is not solved completely. Ad-Hoc On-Demand Distance Vector Routing (AODV) [6] and Dynamic Source Routing (DSR) [7] are examples of reactive protocols. To provide more efficiency and scalability, a third group of protocols was introduced as hybrid routing protocols, which is a combination of both reactive and proactive approaches. However, the above mentioned limitations are still in place. An example of a hybrid routing protocol is Zone Routing Protocol (ZRP) [8]. Finally, a survey and comparison of topology-based protocols can be found in [9, 10].

Position-based or geographic routing approaches were introduced to eliminate some of the limitations of the topology-based protocols in MANETs. These routing protocols rely on having one piece of information and that is the nodes' physical location information. Thus, it is necessary for nodes to obtain their coordinates either by using a *location service* such as GPS or other types of positioning services [11, 12]. A survey of these services can be found in [13]. By employing position information, geographic routing protocols do not need to establish and maintain routes, thereby eliminating routing table construction and maintenance. In this report, we refer to position-based routing protocols generally as geographic routing. The forwarding strategy in these protocols is based on location information of the destination as well as the one-hop neighbours. It is probable that the forwarding scheme fails if there is no one-hop neighbour whose location is closer to the destination than that of the forwarding node. In such cases, *recovery strategies* are introduced to deal with such failures.

The goal in this report is to review different geographic routing protocols as well as the proposed recovery strategies and discuss their suitability for Unmanned Aeronautical Ad hoc Networks.

The remainder of this report is structured as follows: we first present the basic idea of geographic routing protocols and discuss why it sometimes fails. Then, we survey the proposed recovery strategies that are available in the literature. Giving these backgrounds, we subsequently look at different geographic routing protocols proposed for two-dimensional (2D) environments. Finally, we give a general evaluation of the discussed protocols from the perspective of UAANETs and conclude the report.

2. Geographic Routing

One of the early proposed protocols, employing location information was “Location Aided Routing (LAR)” [14]. It is based on DSR but limits the propagation of route request packets to a geographic region where it is most probable for the destination to be located. Therefore, geography is not used for packet forwarding decisions in LAR. It is only used to limit the propagation area. In fact, LAR is classified as a position-based routing protocol. What researchers refer to as “geographic routing” is mainly a solution that employs geographic information for the purpose of routing and data forwarding. Thus, the focus in this report is mainly on such protocols and when we use the term “geographic routing”, we refer to this category of routing protocols.

Geographic routing protocols scale better for ad hoc networks mainly for two reasons: 1) there is no necessity to keep routing tables up-to-date and 2) no need to have a global view of the network topology and its changes. Therefore, geographic routing protocols have attracted a lot of attention in the field of routing protocols for MANETs. These geographic approaches allow routers to be nearly stateless because forwarding decisions are based on location information of the destination and the location information of all one-hop neighbours. Most of these protocols keep state only about the local topology (i.e., neighbours’ location information). No routing table is constructed. As a result, establishment and maintenance of routes are not required, reducing the overhead considerably. Early proposals for geographic routing were based on pure greedy approaches: a packet at an intermediate node is forwarded to the neighbour who is the closest to the destination [15, 16]. Each intermediate node applies this greedy principle until the destination is reached. However, greedy routing does not guarantee delivery even if there is a path from source to destination. The reason being that this principle fails if there is no one-hop neighbour that is closer to the destination than the forwarding node itself. We will introduce the proposed recovery strategies to overcome this problem in Section 2.2.

2.1 Greedy Geographic Forwarding

There are several greedy routing strategies. They can be defined in terms of *progress*, *distance* and *direction* towards the destination. The progress is the distance between a node S and the projection A' of a neighbour node A onto the line connecting S and final destination D (see Figure 1). The larger this distance, the more progress the corresponding neighbour can make. For instance, *the Most Forward within Radius* (MFR) [17] scheme is based on this progress notion.

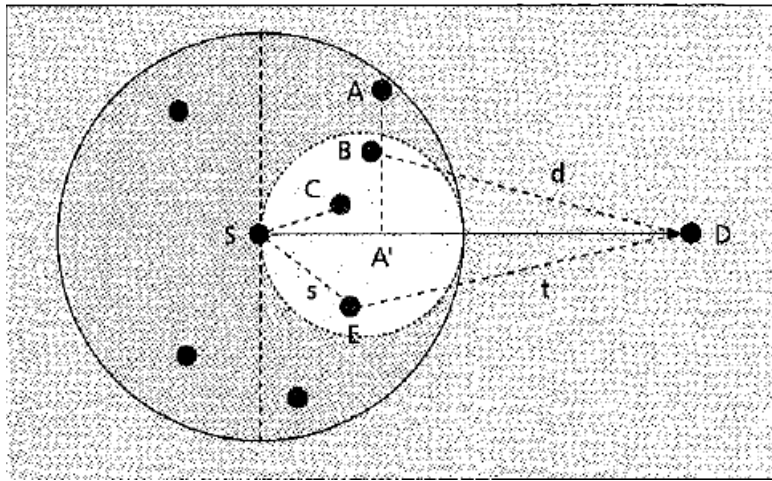


Figure 1. Several greedy routing strategies based on progress, distance and direction [31]

In MFR, the packet destined to destination D is forwarded to the next neighbour who maximizes the progress towards D (e.g., node A in Figure 1). This scheme minimizes the number of hops to reach D . Under this category, there is another scheme called *Nearest with Forward Progress* (NFP) [18], which forwards the packet to the nearest neighbour of the sender that is closer to the destination (node C in Figure 1). It is shown that if all nodes employ NFP, the probability of packet collision is reduced significantly [3]. Therefore, this strategy performs better than MFR. Another greedy strategy, which is widely used, applies the same principle, but uses the notion of distance, and more accurately, the Euclidean distance. That is, an intermediate node forwards the packet to the neighbour with least distance d to the destination, who is closer to D than S (e.g., node B in Figure 1). Direction-based schemes, also called compass routing [19], use the deviation as a criterion. The deviation is defined as the angle between two lines: the line connecting the current node and the next hop, and the line connecting the source and the

destination. The deviation is used to select the neighbour closest in the direction to destination D (e.g., node C in Figure 1.). This scheme aims at minimizing the spatial distance a packet travels.

The main problem with greedy routing is that it does not guarantee delivery to the destination even if there is a path from the source to the destination. This is called a local minimum. An example of this problem can be found in Figure 2. In this example, node S does not have any neighbour in its vicinity (within its transmission range) that is closer to destination D than S itself. However, Figure 2 shows that a valid path from S to D exists. There were several early proposals to overcome this problem such as forwarding the packet to the least backward (negative) progress [20] or simply not to forward such packets and drop them [21]. The problem with the former solution is that looping might occur when there is backward forwarding [3].

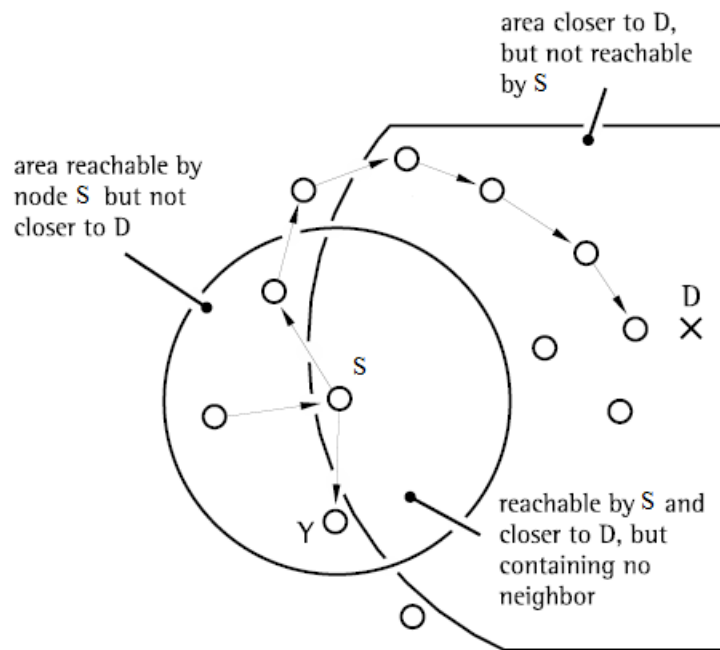


Figure 2. Reaching a dead end (local minimum) [34]

There have also been proposals based on memorization – to keep and use the information about past routing tasks – that guarantee delivery [22]. However, due to increased communication overhead, stateless algorithms based on routing in planar geometric graphs attracted more attention as recovery mechanisms. Therefore, greedy routing is often used in combination with a recovery strategy, which is responsible for handling the packet as long as greedy routing fails. In other words, greedy routing continues until it reaches a local minimum and fails. Then it

switches to the recovery strategy. However, the recovery solution returns to greedy routing after it meets a node that is closer to the destination than the greedy failure node. The return happens either immediately or after some time depending on the type of strategy used. This node may either be the current packet receiver or one of its neighbours. The most prominent recovery strategy is *Face Routing* [23], which is explained in the next section.

2.2 Recovery Strategies

The first geographic routing algorithm based on planar graph traversal that guarantees delivery is *Face Routing* [23]. Face routing attracted a lot of attention as a recovery strategy. First, we review the concept of planar graphs. A geographic planar graph partitions the plane into several inner and one outer face that are bounded by the polygons formed by the edges of the graph. There are no intersecting edges. In other words, the plane is modeled by a two dimensional geometric graph. Each network node v is represented by a point in the plane (corresponding to its location). That means the nodes are vertices. An edge exists between two points u and v if they are close enough to communicate directly (Figure 5). A two dimensional geometric graph is planar if any two edges intersect in their end points only [24]. See Figure 3 for an example of such a planar graph.

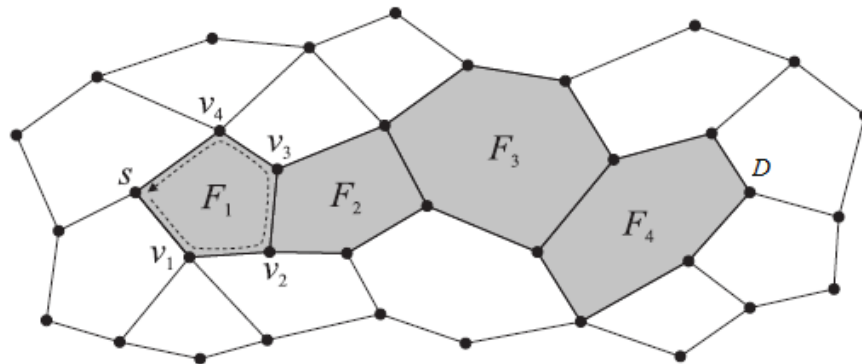


Figure 3. By using left or right hand rule, each packet explores a sequence of faces, making progress towards the destination [24]

However, the graph formed by an ad hoc network is generally not planar; for example in Figure 4, the transmission range of each node contains all the other nodes and there are crossing edges in the network, therefore, the graph is not planar [3]. For this reason, before planar graph routing can be performed, a planar subgraph must be extracted from the complete network graph.

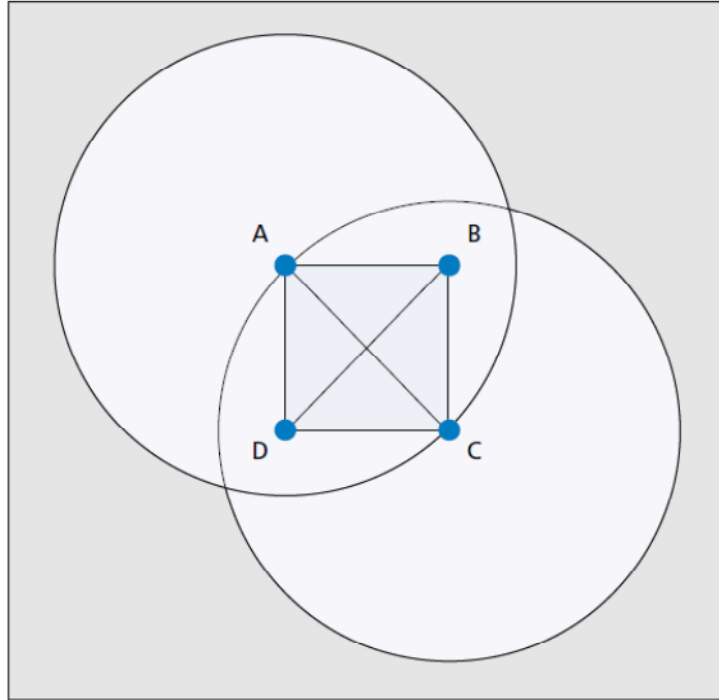


Figure 4. Non-planar graph [19]

In the following, three well-known techniques that are used to locally construct a planar graph are described. By employing these techniques, each node is given a consistent view on the planar graph by removing those outgoing edges that do not comply with the criteria applied to the construction of the planar graph [24].

- *Gabriel Graph (GG)* – A node keeps all outgoing edges uv that form a circle with diameter $|uv|$ that passes through u and v and there is no other neighbour node in between (see Figure 5 (a)) [25].
- *Relative Neighbourhood Graph (RNG)* – A node u keeps its outgoing edge uv if circles with centers u and v and radii $|uv|$ contains no other node than v [26] (see Figure 5 (b)) [25].
- *Localized Delaunay Triangulation (LDT)* [27, 28, 29] – Each node calculates the Delaunay triangulation on its own set of neighbours. The Delaunay triangulation contains all triangles with the following condition: the circle passing through the triangle end points does not contain any other node (See Figure 5 (c)) [25]. Now among all its outgoing Delaunay edges, each node preserves all of those that are preserved by the node on the other edge end point as well.

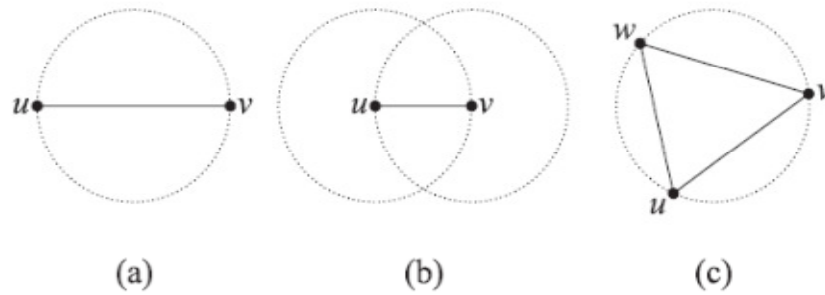


Figure 5. Localized planar graph construction based on (a) Gabriel graph, (b) relative neighbourhood graph, and (c) Delaunay triangulation [24]

Once the planar subgraph is constructed, then the planar graph routing can be performed. The main idea in face algorithm is to forward a packet along the interiors of a sequence of adjacent faces that are intersected by the straight line SD connecting the source node s with the destination node t (in Figure 3), which are providing progress towards the destination node D , e.g. the sequence F_1, F_2, F_3, F_4 depicted in Figure 3 [24]. Face traversal is done in a localized way by applying the well-known right hand rule (or left hand rule): a packet is forwarded along the next edge clockwise (counterclockwise) from the edge where it arrived [3]. For example, applying the right hand rule and starting with the edge s, v_1 in Figure 3 will result in the cycle s, v_1, v_2, v_3, v_4, s . Applying the left hand rule and starting with the edge s, v_4 will result in the reverse cycle s, v_4, v_3, v_2, v_1, s . When the packet arrives at an edge intersecting the line connecting the source and the destination, the next face intersected by this line is handled in the same way. This algorithm is proved to be loop-free and guarantees that a path from the source to the destination will be found if there exists at least one such path in the original non-planar graph [18, 30]. It is important to note that there are variations to face routing. They are discussed in the next section.

2.2.1 Variations to Face Routing

As explained earlier, there is a time in the face routing algorithm to make a decision as to when a face traversal must be interrupted and what new face must be explored next. This decision has been implemented differently by different variations of face routing. Face routing strategies are generally classified into two groups: continuative strategies and volatile strategies. Continuative algorithms keep the line connecting the source and the destination as a reference during the

whole routing process. On the other hand, the volatile algorithms initialize planar graph routing each time a face change occurs [24]. In fact, the node where a face change occurs becomes a planar graph routing start node again. In the rest of this section, we review three continuative strategies followed by three volatile ones. Please note that face routing and its variants are only part of the protocols introduced in the following. In other words, they all start with greedy routing, until the packet reaches a local minimum, where greedy forwarding is no longer possible. Then forwarding is resumed by employing a recovery strategy. Although the point of the following sections is to introduce the variations of face routing, we might also review the greedy part when necessary to get a better view of the entire functionality of the protocol being discussed. Please note that the protocol could also return the packet to greedy mode if any neighbour closer to the destination than the location where a switch to recovery strategy occurred exists. However, for the purpose of focusing on face routing strategies, returning to greedy mode is not shown in the examples given.

Greedy-Face-Greedy (GFG)

In Greedy-Face-Greedy [31], first, face traversal gets started at source node s (see Figure 6) [24]. It is assumed that the left hand rule is applied. As soon as the packet reaches an edge that intersects the source destination line SD at point p , it will change into the face that intersects with the open line segment pD . This happens only if the new intersection point is closer to the destination than the previously encountered intersection point where current face traversal was started. Figure 6 shows an example, where the packet is sent from source node s to destination D . Traversing face $F1$ using the left hand rule, the packet will meet the intersection line SD at point p_1 . This is the first intersection. Now it will change face exploration from $F1$ to $F2$. Again, the left hand rule is applied when traversing face $F2$ until the packet reaches to intersection point p_2 , which is closer to the destination than intersection point p_1 where traversal of face $F2$ was started. In this example, now the packet changes to face $F1$ again, however, only intersection points are considered that are located closer to destination than the last intersection point p_2 . Therefore, intersection point p_3 is the choice and from there, the packet moves to face $F3$. By exploring $F3$ the packet reaches to intersection point p_4 and changes to face $F4$ and finally reaches to the destination node D . Overall, the sequence of visited faces is $F1, F2, F1, F3, F4$.

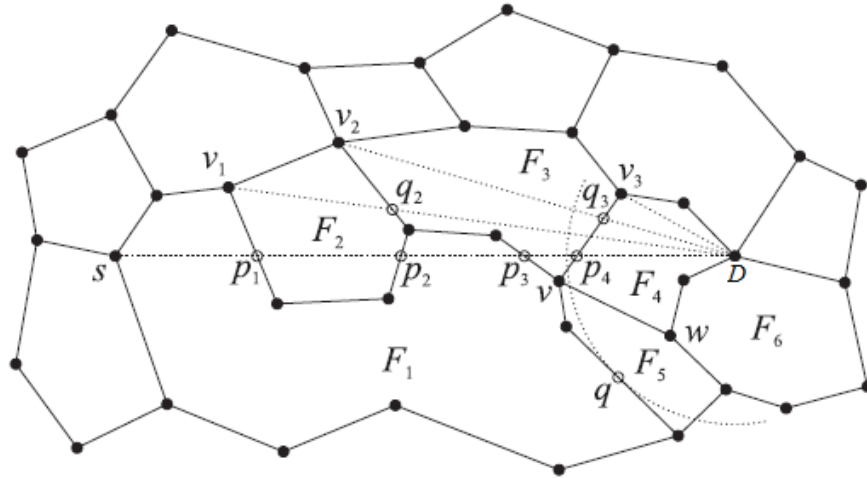


Figure 6. Continuative and volatile strategies [24]

Face -2 (Compass Routing II)

One possible alternative to the previously discussed strategy (i.e., GFG) is to first explore the face completely and find all the intersections with the sD line which connects the source and the destination. Then, the packet will advance to the intersection point, which is closest to the destination among all the encountered intersections during traversal of the current face. This strategy was introduced as Face-2, which was originally called Compass Routing II [32]. For example, in Figure 6, the packet will change its face traversal from face $F1$ to face $F3$ directly because intersection point $p3$ is the closest one to the destination D compared to $p1$ and $p2$. While exploring face $F3$, the packet will find intersection point $p4$ as the closest one to the destination. Therefore, the packet switches to face $F4$ and eventually reaches the destination. As explained, this variant of face routing visits fewer faces in this example compared to the previous strategy (GFG). The sequence of visited faces in this example is $F1, F3, F4$.

Greedy Perimeter Stateless Routing (GPSR)

In fact, GPSR protocol [33] is a complete protocol, consisting of a greedy mode and a recovery strategy called perimeter mode. Here, we focus on the functionality of perimeter mode, as a variant to face routing. Every packet sent using GPSR contains a flag, indicating in which of the following modes it is:

1. Greedy mode
2. Perimeter mode

Initially, all data packets are marked as greedy mode by their source nodes. Once the greedy routing fails, the packet is marked as being in perimeter mode.

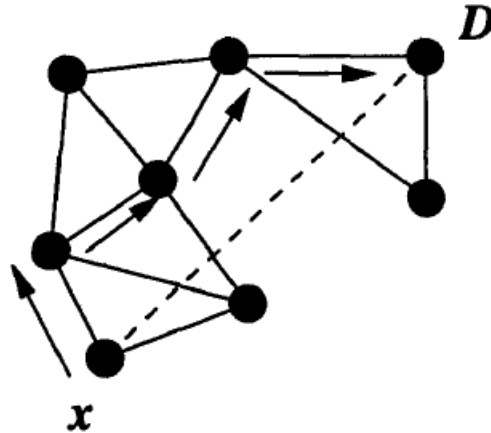


Figure 7. Perimeter Forwarding Example [33]

In perimeter mode, GPSR too, performs a simple planar graph traversal by employing the right hand rule (the same is possible for the left hand rule as well [24]). Suppose the mode changes to perimeter at node x for a packet destined to D (see Figure 7). From here on, the packet is forwarded by employing the right hand rule, traversing the faces intersecting the line xD . On each face, the traversal continues until the packet reaches to an edge that crosses line xD . At that edge, the packet moves to an adjacent edge, the first edge of which is determined by simply choosing the edge lying in counterclockwise direction from the intersected edge. Thereafter, as mentioned, the packet is forwarded around that face using the right hand rule. Figure 7 shows an example of perimeter forwarding starting at node x if it would continue all the way to the destination D . However, as mentioned earlier, true GPSR forwards greedily when neighbours closer to destination are available. The sequence of edges traversed by the right hand rule is called a *perimeter* [33].

In order to determine if GPSR can return to greedy mode, a field called L_p is considered in the packet header, which records the location where the packet entered into perimeter mode (i.e., the location where greedy forwarding failed). This location is used at subsequent hops to determine whether the packet can be returned to greedy mode. That is done by comparing L_p with the

location of the current forwarding node. GPSR returns the packet to greedy mode if the distance between the forwarding node and the destination D is less than that from Lp to D .

GPSR considers the case when the destination D is not reachable. That is, node x (the location where GPSR entered perimeter mode) and destination D are NOT connected by the graph. GPSR's solution to detect such cases is as follows: the disconnected node lies either inside an interior face, or outside the exterior face. The packet will eventually reach this face and will traverse it completely, without finding any intersection point with the line xD , which is closer to destination D than the location where packet entered current face. Then the packet traverses the first edge it took on this face for the second time. In order to notice the repetition of forwarding on this edge, GPSR employs another field in the packet header (stored for such cases) which is called eo . This field records the first edge traversed on the current face. When the packet traverses the first edge it took on this face for the second time, eo shows that it is the second time the packet is forwarded on edge eo and GPSR drops the packet as the destination is unreachable.

Greedy Other Adaptive Face Routing (GOAFR and GOAFR+)

GOAFR (pronounced as “gopher” [34]) also starts routing by performing greedy forwarding and switching to a variation of face routing when reaching to a local minimum with respect to the distance of the current node (none of whose neighbours is located closer to the destination) from the destination. The face routing technique that GOAFR employs has two major differences compared to the traditional face routing. GOAFR also explores the boundaries of a face by employing the right hand rule; however, the algorithm is adaptive in doing so by restricting face traversal to a searchable area (an ellipse or a circle), which will be resized during algorithm execution such that it contains a complete optimal path. This restriction is applied with the goal of staying competitive with the shortest path between the source and the destination. In other words, this is to bind the cost of the algorithm by the cost of an optimal path between the source and the destination. The second difference is that it advances to the node on the current face boundary which is closest to the destination. This is different from the traditional face routing in which face switching occurs once an intersection point on the line between the source and the destination is reached. This version of face routing was originally called Adaptive Face Routing (AFR) by the authors. GOAFR aims to make the protocol asymptotically optimal with respect to its execution cost compared with the cost of the optimal path (shortest path) [34]. The authors

have studied this issue and have found that, once in face routing mode, an algorithm is required to explore the complete boundary of the current face in order to be asymptotically optimal [35]. Therefore, the authors introduced an improved version of GOAFR called GOAFR+, which does not return to greedy forwarding immediately after recovery. However, the algorithm is designed in such a way that it does not have to explore the complete face boundary in face routing mode either and yet does conserve asymptotic optimality. The algorithm returns to greedy mode as soon as a certain return condition holds as explained in the following. To review a more complete version of this protocol, we proceed with discussing GOAFR+.

Before the algorithm starts, three parameters are chosen, which remain constant throughout the execution: ρ_0 , ρ , and σ . Parameters ρ_0 and ρ are used in defining and adjusting the restricted searchable area. Parameter σ along with two counters p and q are used to define the condition for returning to greedy mode. The following conditions hold for the parameters: $1 \leq \rho_0 < \rho$ and $0 < \sigma$. The simulations done by the authors of GOAFR+ revealed that the following values are good choices for practical purposes: $\rho_0 = 1.4$, $\rho = \sqrt{2}$, and $\sigma = 1/100$ [35]. Counter p keeps track of how many of the nodes visited during the current face routing phase are closer to the destination and q keeps track of those that are not closer to the destination. The comparison is made against the location of the node where the local minimum occurred [34].

The algorithm starts by defining a circle C centered at destination D with radius $rc = \rho_0 |SD|$. Then it proceeds by taking greedy steps until the packet either reaches to the destination D or a local minimum. In the former case, the algorithm terminates, in the latter case it will switch to face routing as described below. Also, whenever possible, the algorithm reduces C 's radius ($rc = rc/\rho$) as long as the currently visited node stays within C . Now suppose greedy forwarding fails at node u and the algorithm enters into face routing. It will start exploring the boundary of face F , which contains the connecting line uD immediately reachable from u . During exploration of F the two counters p and q are updated. In addition, the following conditions are checked and actions are taken accordingly:

1. If circle C is hit for the first time, turn back and continue exploring F 's boundary in the opposite direction.

2. If circle C is hit for the second time, the nodes visited so far are checked. If none is located closer to the destination than u (the node that entered into face routing), then the algorithm enlarges circle C ($rc = \rho rc$) and continues exploring face F as if started from u . However, if one of the nodes visited so far is closer to the destination, it advances to that node and continues with greedy mode.
3. If $p > \sigma q$, it means we have visited more nodes on F 's boundary closer to the destination than nodes not closer to the destination. In this case, the packet advances to the closest node seen so far (if it is not the currently visited node) and then the algorithm continues in greedy mode.

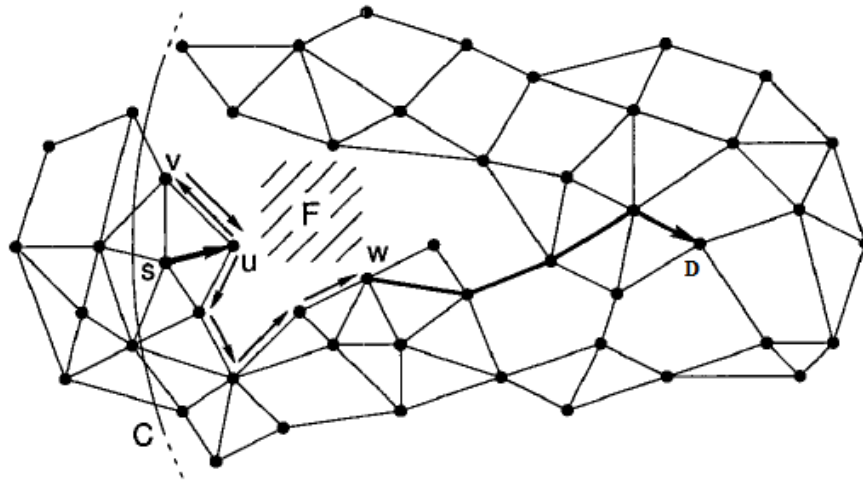


Figure 8. An example of GOAFR+ algorithm [34]

Figure 8 shows an example of GOAFR+ in which the algorithm first starts from node s in greedy mode. It reaches to node u , which doesn't have any neighbour closer than itself to the destination (local minimum). Here, the algorithm switches to face routing mode and starts exploring the boundaries of face F . At node v , the algorithm hits the bounding circle C and thus turns back to continue the exploration of F in the opposite direction. Counters p and q are updated after each step. At node w , the number of nodes visited so far that are closer to the destination D is two; that is, $p = 2$. And the number of nodes that are not closer to the destination is four; that is, $q = 4$. Assuming that σ is set to be $1/4 \leq \sigma \leq 1/2$, the fallback condition becomes true: $p > \sigma q$. Therefore, the algorithms returns to greedy mode and continues until it finally reaches to destination D .

In their primary proposal, the authors of GOAFR+ have defined an ellipse instead of a circle, in which the principles and discussed steps are the same and resizing the ellipse is done by changing the length of the ellipse's major axis.

Greedy Path Vector Face Routing (GPVFR)

All the geographic routing protocols discussed up to here in this survey aim to remain memory-less; that is, the goal is to not keep any routing information except for local neighbour location information. Each node receives information of only its neighbours (i.e., nodes that are one hop away). However, the idea in Greedy Path Vector Face Routing (GPVFR) [36] is to maintain and make use of a small amount of local face information at each node. The authors of GPVFR believe that having information about only the immediate neighbours is often insufficient for the node to make a good decision on the forwarding direction when local minimum occurs and the algorithm switches to face routing. Therefore, GPVFR is based on employing more information about planar graph. First the authors show that there exists an oblivious (memory-less) algorithm, Oblivious Path Vector Face Routing (OPVFR), that guarantees packet delivery in any planar graph if nodes have complete face information [36]. However, in practice, the assumption that nodes can maintain complete face information is impractical. Therefore, the authors introduced two algorithms to overcome this problem:

- *Path Vector Exchange (PVEX)*, which is a practical asynchronous distributed algorithm to propagate and maintain local face information efficiently. It also keeps track of network membership changes.
- *Greedy Path Vector Face Routing (GPVFR)*, which is a non-oblivious algorithm that guarantees packet delivery even when nodes do not have complete face information.

In the following, we review all three algorithms, *Path Vector Exchange (PVEX)*, *Oblivious Path Vector Face Routing (OPVFR)* (the two used in constructing the final *GPVFR*) and *Greedy Path Vector Face Routing (GPVFR)* itself.

Path Vector Exchange (PVEX): As mentioned earlier, this algorithm is used to collect and share local face information. Nodes periodically send beacons to their neighbouring nodes to inform

them of their position and face information. The information to be exchanged is organized as follows [36]:

- F_v : An indexed set of the faces adjacent to v ,
- S_f : list of successor nodes for face $f \in F_v$,
- P_f : list of predecessor nodes for face $f \in F_v$, and
- Seq_f : sequence number for face $f \in F_v$ (an integer).

The number of faces of a node is exactly equal to its degree in the planar graph. To get local face information, each node processes only messages that come from nodes with which it shares a face f and where the received sequence number for f is greater than or equal to the local sequence number for f . Then the node updates its state accordingly. The sequence number Seq_f for a face f is incremented in two cases: (i) when a node detects that its successor on f has left the network, or (ii) when a new node joins the network. S_f and P_f are truncated to a maximum length h , which is a system parameter. This is to limit the length of the path vectors propagated in PVEX to some constant h . This results in knowledge of only those nodes that are up to $h + 1$ hops away. At each node, this protocol stabilizes locally when the successor list intersects with the predecessor list and it has obtained full information for a face. This happens as soon as the sequence number no longer increases for several inter-beacon intervals.

Oblivious Path Vector Face Routing (OPVFR): As mentioned before, in this algorithm nodes have full face information, which can be obtained and maintained by running PVEX with $h = \infty$ (introduced in the previous part). Since complete face information is available at each node, the node receiving a packet destined for the destination finds the edge among the set of all edges on its adjacent faces that has the minimum Euclidean distance to the destination. Having an edge (x,y) , the minimum Euclidean distance of it to destination D is the length of the perpendicular if the segment \overline{xy} contains the projection of D on it; otherwise, it is $\min(|xD|, |yD|)$. This closest node to destination D on edge (x,y) is called the target node. To summarize, the forwarding rule in the OPVER protocol consists of two modes:

1. **Direct.** If destination D is located on face f that the packet holder is located on, then forward the packet to any neighbour on f (containing both the packet holder and

the destination D) whose distance to D is less than the distance of packet holder to D . Otherwise, follow step 2.

2. **Target node:** Forward the packet to any neighbour on face f that contains the target node, such that the distance of that node to the target node is less than the distance of the packet holder node to the target node. The distances are considered along a path on face f .

Greedy Path Vector Face Routing (GPVFR): As opposed to the OPVER algorithm, nodes are not required to have complete face information in this algorithm and the authors refer to this protocol as a non-oblivious routing protocol. Employing the previously discussed algorithms, the GPVER protocol consists of the following three modes:

- Greedy: using neighbour information, it performs greedy forwarding,
- OPVER: using face information, it performs greedy forwarding, and
- Perimeter: perimeter traversal (as discussed in GPSR).

In fact, the main difference between GPVER and other geographic protocols discussed so far is that GPVER takes advantage of the OPVER algorithm after greedy forwarding fails (reaching a local minimum) and before switching to face routing. In other words, it does not switch to face routing once greedy forwarding fails, but switches to another protocol OPVER, yet based on face routing and more similar to greedy forwarding. OPVER acts like greedy forwarding with the difference that nodes have longer horizon and the algorithm is restricted to forwarding on the planar faces (edges). However, in some topologies, OPVER also might fail and that is when GPVER proceeds with face routing. The face routing employed in GPVER is slightly different from the previously discussed ones. The packet is forwarded along the edges of the planar face that contain the line from the *node* to the destination (and not necessarily the line connecting the *source* to the destination). In addition, instead of using an arbitrary right hand rule, the direction to start traversing the face is based on the currently known set of path vectors. The idea is to make face traversal more efficient by using the available face information.

Figure 9(a) shows an example of switching to OPVER mode at node s . Using the available information, node s finds the target node n^* . Based on the chosen performance metric to be

either hop count or Euclidean path distance, node s forwards the packet to either k_1 or k_2 respectively.

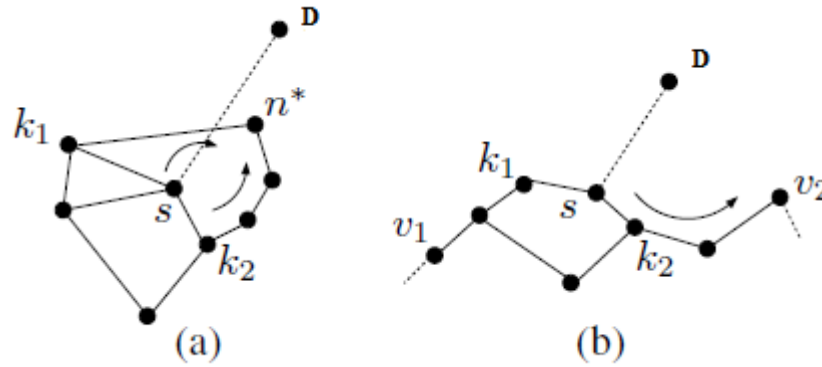


Figure 9. Examples of path selection for GPVER [36]

Figure 9 (b) is an example of the case when node s can neither proceed with greedy forwarding nor OPVER protocol. Therefore, face routing is started at node s and having performed limited-length path vector exchange (assuming s has information of nodes up to 3 hops away) it forwards the packet to k_2 as it is closer to the destination D . The forwarding direction and mode must be recorded in the packet so that the next node will be able to determine the next edge of the face being traversed.

Overall, when switching to perimeter (face routing) mode, the current node finds the face containing the line segment between the node starting face routing and the destination. Then using the face information, it forwards the packet to the neighbour that is closer to the destination. This will determine if the face will be traversed clockwise or counterclockwise. The forwarding direction c is tagged into the packet. In addition, the first edge traversed on the current edge as well as the node that started face routing (called anchor node) are recorded in the packet. A change in the face occurs when the packet reaches a node that has an outgoing edge, which intersects the line segment between the anchor node and the destination. In this case, the new node is set as the anchor node and all the above mentioned actions are performed. If the next edge to be traversed on the current face f is already recorded in the packet, it means a loop has occurred and there is no path to the destination and thus the packet is undeliverable.

The authors have found that this mechanism results in a better performance compared to other geographic routing protocols. They discuss that the cost of this improved performance is a small amount of additional storage and yet, the required bandwidth is still comparable to other geographic routing protocols such as GPSR and GOAFR+ [36].

2.2.2 Alternatives to Face Routing

As discussed in the previous section, there are different types of face routing strategies and as the reader may have noticed, they are all based on planar graph. Although face routing and its variations have attracted a significant research effort, there are also other recovery strategies proposed in the literature that use other mechanisms than face routing. In general, recovery strategies, also called void handling techniques, can be divided into six categories based on the techniques used [37]:

- Planar graph-based
- Geometric
- Flooding-based
- Cost-based
- Heuristic
- Hybrid

Planar graph-based void handling: All of the protocols discussed in Section 2.2.1 belong to the first category. They use face routing and its variations to tackle the problem of greedy forwarding failure. In addition, there are two other protocols that offer recovery strategies based on planar graphs. Beacon-Less Routing protocol (BLR) [39] uses Request-Response technique. In fact, the major difference in BLR is that the nodes do not have neighbours' location information first but acquire the information when needed. Priority-based Stateless Geo-Routing (PSGR) [39] uses Bypass technique as void handling techniques.

Request-Response: BLR [38] is different in that it does not construct a complete planar subgraph for the original network graph in advance. However, a partial planar subgraph is constructed on demand and only for nodes around the void node (i.e., the node at which communication voids with respect to the destination occurs). The void node first broadcasts a request. It is only at this

time that the neighbouring nodes reply by sending their position. After receiving the location information from all its neighbours, the void node extracts a planar subgraph using a localized planarization algorithm such as Gabriel Graph (introduced in Section 2.2) and then forwards the packet by the right hand rule and returns to greedy forwarding when it finds a node closer to the destination as discussed before. This technique increases overall network delay as the void node sends an instant request for the information of the neighbouring nodes [37].

Geometric void handling: In this approach, the geometric properties of nodes are exploited to identify topological structures called holes. BOUNDHOLE [40] was the first algorithm to work on holes. In [40] holes are defined as the areas of the network encompassing all the nodes in the network that can possibly become void nodes in greedy forwarding. By doing so, we can conclude that any packet that gets stuck at a void node is located at the boundary of at least one of the holes. As opposed to planar graph approaches, computing and storing information about holes for those parts of the network that can become problematic is necessary. Then a path can be found either on demand (i.e., when greedy forwarding fails) or the paths can be discovered in advance and stored locally along the boundaries of holes. The BOUNDHOLE protocol [40] uses a rule called TENT rule (applied by the void node itself) to identify a hole around a void node. Here is a brief explanation how the TENT rule works. As Figure 10 shows, for each node p , all its one-hop neighbouring are ordered. For each pair of adjacent neighbours of p , that is u and v in this example, the intersection of the perpendicular bisectors of up and vp is found. If the intersection (point O in his example) falls within the transmission range of p , the black region must also lie within the transmission range of p . In this case, node p will not get stuck for any destination in the direction of upv . However, if O is outside, there is the possibility that node p will become a void node for some destinations. Once node p determines that it might become a void node for some of the destinations (using the TENT rule), it starts BOUNDHOLE algorithm to bound a hole around it. The basic idea in this algorithm is to use a packet to sweep over the stuck direction and mark the boundary of the hole. In the example of Figure 11, the packet is sent from p to its neighbouring node $t1$ in a counterclockwise order. Then it is passed to the next neighbouring node, $t2$ and thus bounding the hole. This process is repeated until the packet marks the boundary of the hole and returns to node p . The boundary of the hole is cached locally and thus can give guidance for the subsequent stuck packets being routed around the void area.

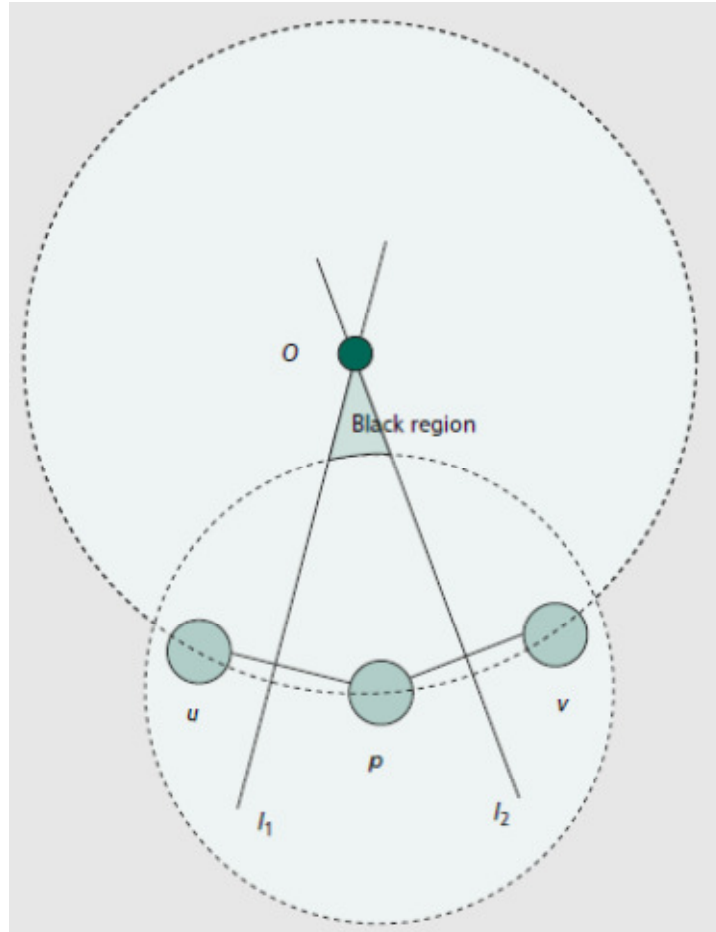


Figure 10. The TENT rule to detect void nodes [37]

That is, once the void node receives a packet, it knows it is located on the boundary of a hole and that there are other nodes (either void or non-void) forming this boundary. It then routes the packet along the boundary of the hole. Figure 12 from [37] shows an example, where node p routes the packet along the boundary. Once it reaches to a node closer to the destination q (e.g., either u or v) the algorithm can switch back to greedy forwarding.

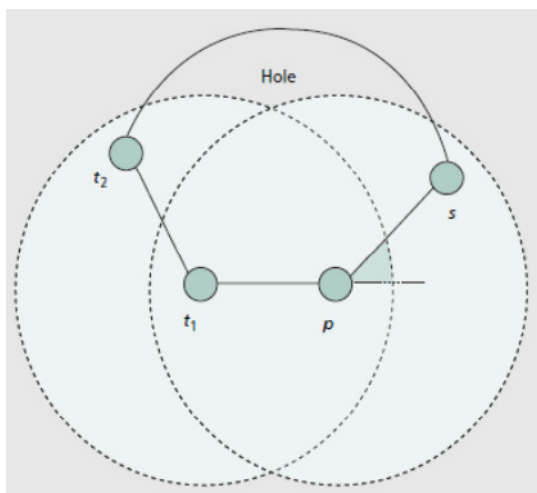


Figure 11. An example of BOUNDHOLE algorithm, building the boundary of a hole [37]

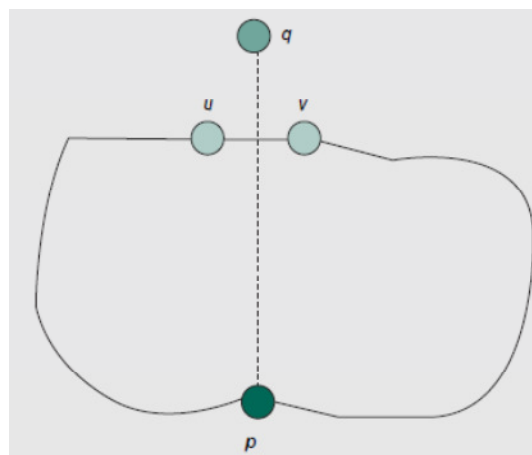


Figure 12. An example of routing around a hole

Flooding-based void handling: The idea in this category is to make use of flooding, the simple form of communication, when a packet gets stuck at a void node. However, as it is known, full flooding is costly and inefficient in terms of resource utilization. In addition, in geographic routing only one destination is to receive the packet that is stuck at void node. Therefore, the proposed algorithms have employed restricted flooding with the aim of controlling the range of flooding as well as the frequency of its occurrence at void nodes. This mechanism is also called partial flooding. There are three of these protocols in the literature that are reviewed below.

One-hop Flooding: As the name suggests, in this algorithm the void node broadcasts the packet only to its one-hop neighbouring nodes. The void node also records the packet ID to be used if it receives the packet again from its neighbours. Each neighbouring node acts independently and forwards a copy of the packet by greedy forwarding. If the neighbouring node has to select the void node in greedy forwarding, the void node sends a rejection packet back to the neighbour so that the neighbouring node will select the next best node from its neighbours. If it fails to find any appropriate node, the neighbouring node itself becomes a void node. This process can be better seen in Figure 13. Node *S*, being a void node, initiates one-hop flooding. Neighbours *A* and *B* receive the packet (Figure 13 (a)). A copy of the packet is delivered to destination *D* via node *A* by greedy forwarding in Figure 13 (b). Neighbour *B* however, sends a copy of the packet to its neighbour *S* but receives a rejection packet from *S* (Figure 13 (c)). The purpose of the rejection packet is to inform *B* that the packet has no way to go towards the destination and thus

B must proceed with another neighbour if any, or conclude that it is a void node too. Since B cannot locate any other neighbour closer to the destination, it becomes a new void node. Therefore, it initiates a one-hop flooding, which will turn into greedy forwarding at node C . It is via C that a copy of the packet is delivered to D (Figure 13 (d)).

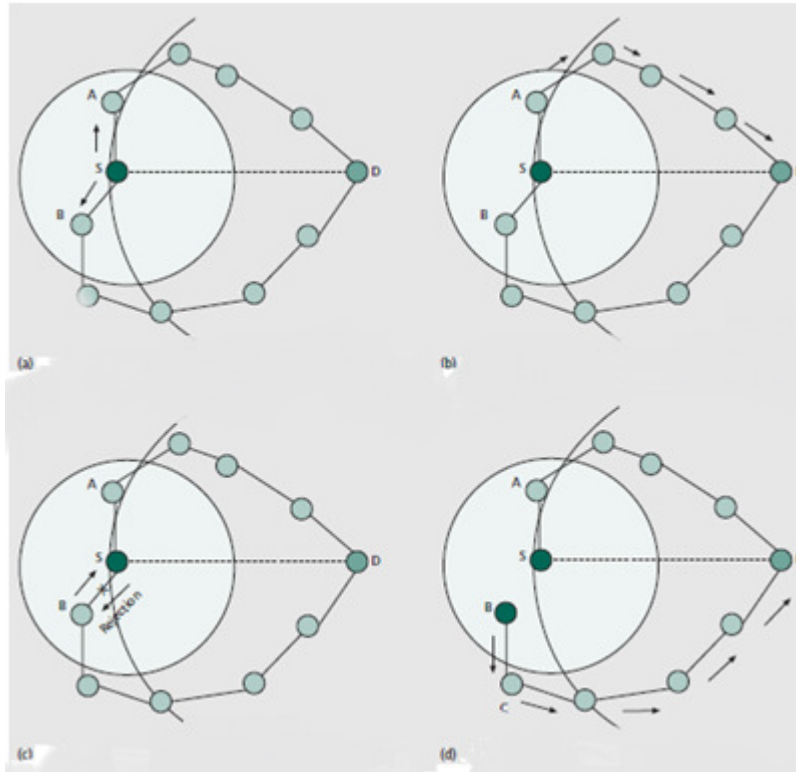


Figure 13. An example of one-hop flooding to handle a void [37]

Partial Hop-by-hop Routing: This strategy is proposed in the Geographic Routing Algorithm (GRA) [41]. Once a packet reaches a void node, a route discovery will be initiated on demand to find a path from void node to the destination. As in other route discoveries, routing tables will be updated at all nodes on the path from the void node to the destination. Once this step is completed, the packet can be routed from the void node to the destination by partial hop-by-hop routing. Flooding is used in the route discovery phase. Please note that another class of route discovery mechanisms (other than flooding-based techniques) have been proposed for route discovery such as the depth first-search based technique in [42]. However, discussion on such protocols is out of the scope of this survey.

Partial Source Routing: PSR [43] is an on-demand scheme in which a void node will forward the stuck packet to either the targeted destination node directly or indirectly to a node that has less distance to the destination than itself. PSR is performed in two phases: partial path discovery and source packet forwarding. In the first phase, the void node tries to find a partial path by flooding a discovery packet to all of its two-hop neighbouring nodes. If the discovery fails, the void node extends the flooding range of the discovery packet by sending it another time to its three-hop neighbours. This process is repeated until the target node is found or the maximum number of runs for path discovery is reached, in which case the stuck packet will be discarded.

In the source packet forwarding phase, the void node knows at least one path to the desired destination or to another node that is closer to the destination than itself. Now the source node includes this path in the packet and sends it to the next hop specified in the path. Figure 14 demonstrates how PSR works. Node *V*, being a void node, initiates a partial path request packet to its two-hop neighbouring nodes (Figure 14 (a)). Since this path discovery fails, node *V* sends another partial path request packet, this time to its three-hop neighbouring nodes (Figure 14 (b)). A partial path reply packet is sent back from node *C*, which is closer to the destination (Figure 14 (c)). Now the data packet, which was stuck at node *V*, will be forwarded to node *C* by source packet forwarding. Then greedy forwarding is resumed.

Cost-Based Void Handling: The idea in this category of void handling techniques is to resolve the problem of voids based on the cost assigned to the nodes in the network. The definition of cost varies in different contexts. In cost-based void handling, a packet flows from a node with a higher cost to a node with a lower cost [44]. The void node at which a packet gets stuck changes the value of its cost to a value larger than the value of the cost originally assigned to it. For example, if the cost is defined to be the Euclidean distance to the destination, the void node increases its cost to a value larger than its Euclidean distance. This will lead the packet from high cost nodes to low cost nodes to get around the void. Two main protocols of this category are cost-based forwarding in Partial-partition Avoiding Geographic Routing-Mobile (PAGER-M) [45] and Distance Upgrading Algorithm (DUA) [46].

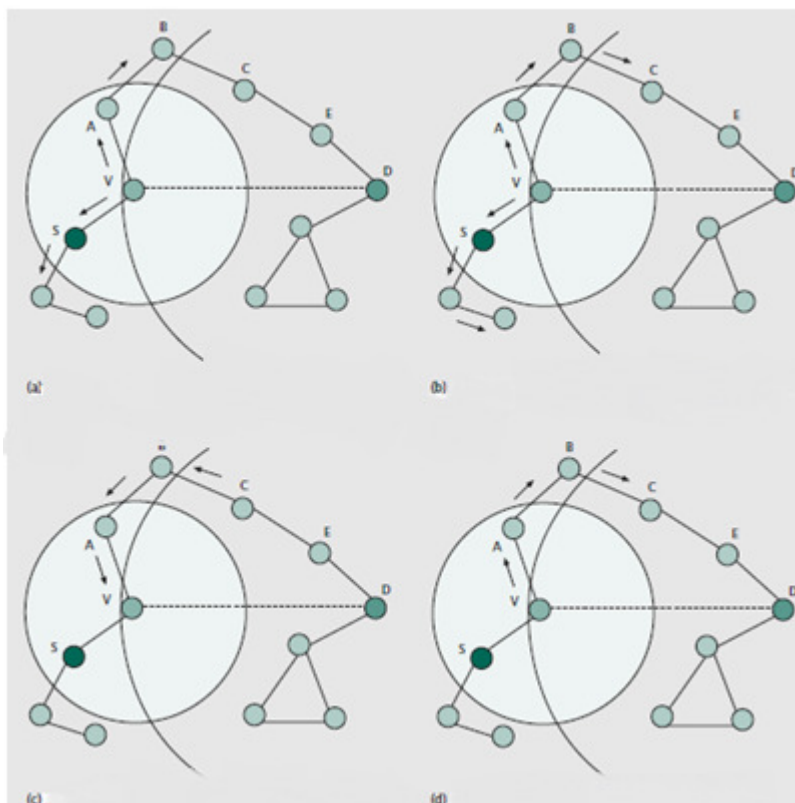


Figure 14. An example of Partial Source Routing to handle a void [37]

Cost-based Forwarding: The cost-based forwarding approach used in PAGER-M [45] consists of two phases: the shadow-spread phase and the cost-spread phase. Shadow nodes are referred to as void nodes and potential void nodes. Nodes that are non-void are referred to as bright nodes. This information, along with the position information, is exchanged during the first phase among the one-hop neighbouring nodes. Therefore, this process divides the network into shadow areas and bright areas. Figure 15 (a) shows an example of this phase [37]. Then, in phase two, as in the example of Figure 15 (b), node A, which is identified as a void node, finds out that all its neighbouring nodes have larger costs. Therefore, node A increases its cost to a value larger than the maximum cost of its neighbouring nodes by a constant δ (set to 3 in the example of Figure 15). Now nodes B and C adjust their costs accordingly to 23 and 25 respectively (Figure 15 (c)). This process ends when node A increases its cost once more to 28 (Figure 15 (d)). After this phase, the shadow nodes (A, B and C in this example) have at least a neighbouring node with a smaller cost. As Figure 15 (e) shows, any stuck packet at void node A can now get out of void nodes by cost-based forwarding.

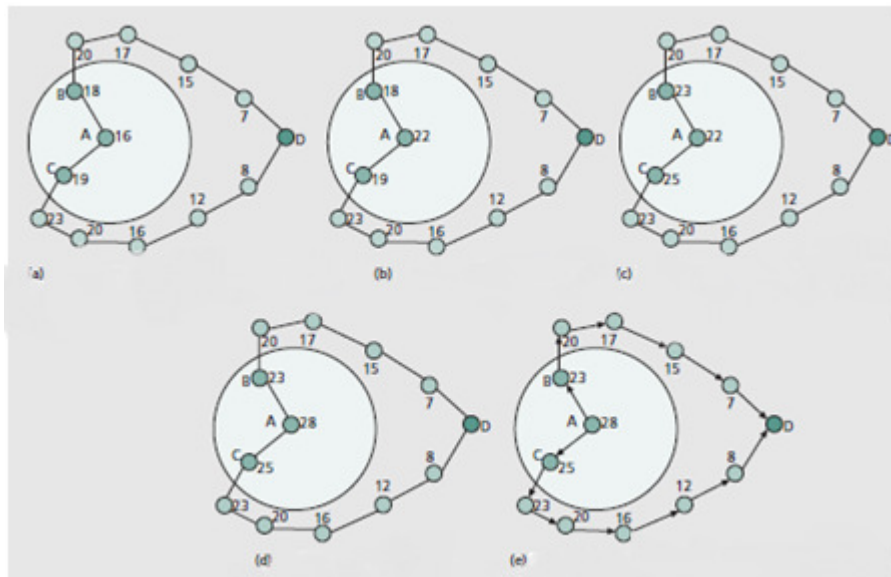


Figure 15. An example of cost-based void handling technique used by PAGER-M

Distance Upgrading Algorithm: DUA [46] investigates possible problems that might occur in cost-based void handling technique used by DUA. Three different cases of distance upgrade are employed in DUA (not discussed in detail in this survey). For cost upgrade in PAGER-M, δ is set to 100, defined as the diameter of the area where the sensors of a sensor network are deployed [46]. The example in Figure 16 helps explain one of the problems and its solution in DUA. Figure 16 (a) shows the original cost (Euclidean distance to the destination) distribution. The first problem is that DUA may produce inefficient routing paths after distance upgrade as shown in Figure 16 (b). That is because nodes X finds itself as a void node and increases its cost by δ (i.e., 100 in this protocol). The neighbouring node V now performs a distance upgrade by choosing a distance value just under that of X . Subsequently, nodes W and Z set their distance values just under that of V . In the resulting cost distribution, most sensors follow an inefficient path from X clockwise around the void to reach the destination. That is because the cost for node X is raised too high. To fix this problem, DUA proposed a distance downgrading mechanism (explained in detail in [46]) to bring the cost down to an appropriate level as shown in Figure 16 (c).

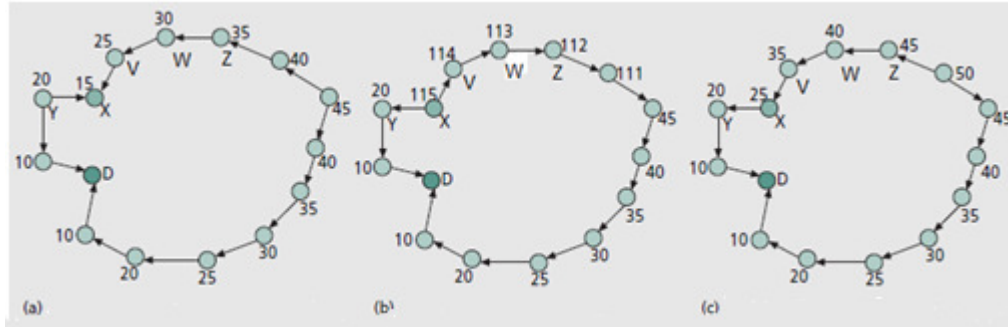


Figure 16. An example of distance downgrade procedure in DUA [46]

Heuristic Void Handling: A number of protocols based on intuitive ideas are proposed to overcome the problem of voids. In the following, we review some of them in this category.

Alternate Network: This is a costly solution. The idea is to exploit an alternate network such as a satellite network if it is available. Stuck packets can get around the voids by using the alternate network as intermediate hops. The nodes in the network must also be equipped with the necessary additional wireless networking techniques [47].

Active Exploration: In this approach the void node increases its transmission power in order to increase its transmission range so that a node with positive progress is reached. This active exploration technique was proposed in [48]. The node must be able to adjust its power. One possible problem is that this action increases the interference to the neighbouring nodes, which can affect the performance of the MAC layer.

Passive Participation: Once a node identifies itself as a void node, it discards the packet and does not forward the subsequent packets. The node will periodically check if it has a new neighbouring node with positive progress so that it can resume participation in packet forwarding towards the destination. The intermediate nodes will eventually learn about the absence of this particular node (void node) and find other possible paths. Consequently, this results in avoiding the void nodes in some bad zones. In the example of Figure 17 [37], node V is a void node, at which the packet gets stuck. Node V drops the packet. The subsequent packets will be routed to the second best positioned node A and will reach the destination via path ABCE [49, 50].

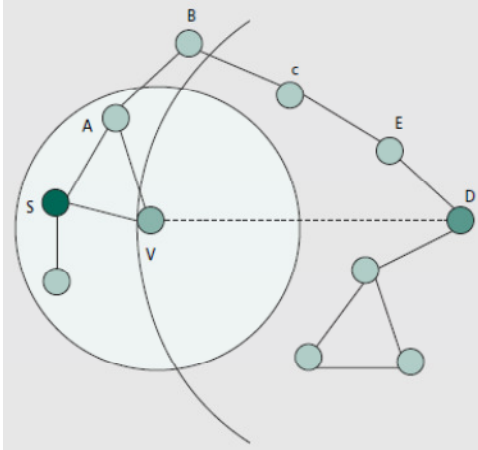


Figure 17. Passive participation – example

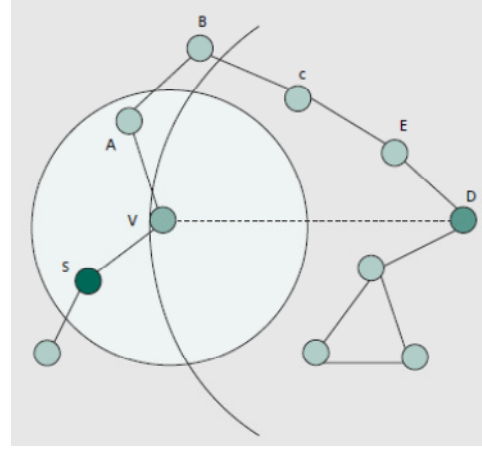


Figure 18. Failure of passive participation - example

Depending on the topology of the network, passive participation can be very ineffective. For example, in Figure 18 [37], after node V becomes passive, node S does not find another neighbour with a positive progress and thus the subsequent packets will all fail to reach to the destination while a valid path from source S to the destination D exists: $S-V-A-B-C-E-D$. It is shown in [48] that the passive participation strategy is not efficient in a wireless network with low node density, where the nodes are deployed randomly.

Void Avoidance: As opposed to the passive participation mechanism, a void node actively sends a packet to its upstream node to announce the existence of void. For example in Figure 19 [37], node V sends a back pressure packet to its upstream node I to inform it that node D is not reachable from it. Node I consequently sets the delay to send to node V to infinity, stops sending any packet to node V and uses node A as next hop instead. If node A does not exist, back pressure will continue until a new path is found.

Intermediate Node Forwarding (INF): This is a probabilistic approach using the intermediate geographic locations [52]. A void node discards the packet and sends a NAK packet back to the source node. The source node then defines a circle around the midpoint of the line connecting the source and the destination and chooses a single intermediate position randomly in that circle. A new copy of the packet now has to traverse that intermediate position. If the packet is discarded again, the radius of the circle is increased and another random position is selected. This procedure will continue until

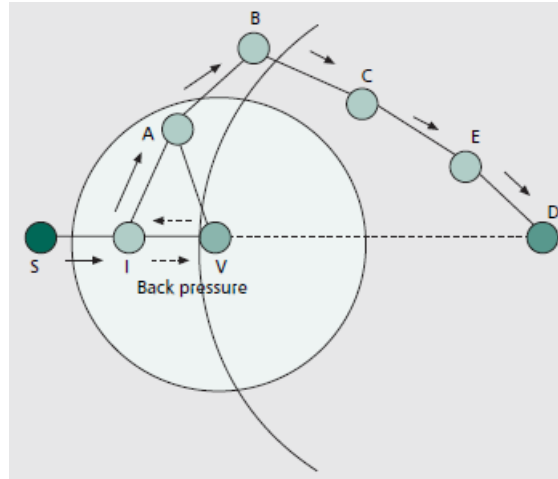


Figure 19. An example of void avoidance

the packet reaches to the destination or until a predefined value is reached and the source assumes that the destination is unreachable. Figure 20 from [52] is an example of INF. The source is node A and the destination is node G. The packet gets stuck at node C, and then gets dropped and a NAK packet is sent back to source node A. Node A then defines a circle with radius r_1 , selects L_1 randomly as the intermediate location. A new copy of the packet traverses AB and then gets out of INF mode at node C because C is close enough to L_1 . However, the packet gets dropped again. This time the source node A increases the radius to r_2 , chooses another intermediate location L_2 and sends a new copy of the packet, which this time passes

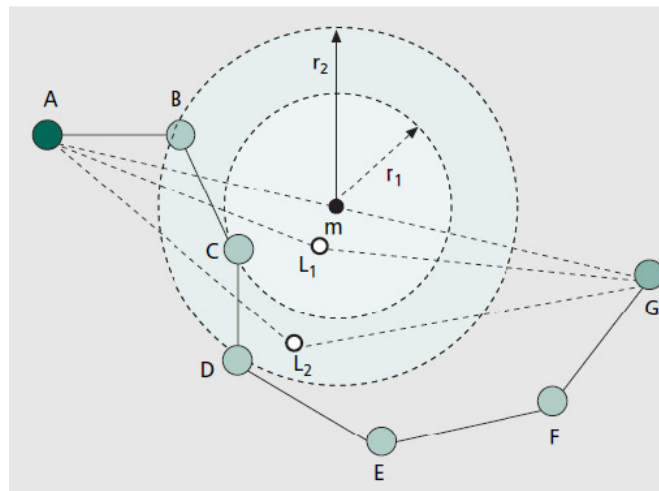


Figure 20. An example of Intermediate Node Forwarding

through node C to D . The packet will then be delivered to the destination G via E and F using greedy forwarding.

Anchored Geodesic Packet Forwarding (AGPF): This technique is different in that it is a preventive approach, trying to prevent a packet from encountering voids even before the packet is sent. AGPF [53] uses loose source routing information, which is included in the packet header. Anchors are geographical locations, computed by source nodes using the path discovery methods presented in [53]. Between the anchors, the packet is greedily forwarded in the direction of the anchors towards the destination. It is highly probable that the packet will reach the destination if the anchors are set accurately.

Hybrid Void Handling: The solutions in the hybrid void handling category combine two or more techniques from either the same category or different categories. There are mainly two reasons to look at hybrid solutions. One is that one single void handling technique might not be capable of handling the voids effectively for different network topologies. An example is AGPF (introduced above), which is a preventive technique. However, it might fail to prevent the occurrence of voids in advance. Therefore, it uses perimeter routing in cases that a void happens. The other reason to have hybrid techniques is to enhance the efficiency of the void handling technique. In the following, four of these hybrid solutions are reviewed.

Partial Source Routing plus Passive Participation: This approach is mainly useful for wireless networks where energy resources are scarce, such as sensor networks. If the energy level at a void node is lower than a threshold value, and the node learns by a mechanism that its own data is more critical, it can switch to passive participation mode. By not forwarding other data packets, it reserves the energy for sending its own packets. In this hybrid approach, even if such a node receives a partial path request packet, it does not respond with a path reply packet if it is passive for that destination [54].

BOUNDHOLE plus Restricted Flooding: The BOUNDHOLE proposal, introduced earlier in this section fails in void handling if the destination is inside the hole. For example, in Figure 21 [37], void node p is closer to destination q than other nodes on the boundary of the hole. Destination q is reachable via nodes v' and u' . However, these nodes are located outside and inside the hole respectively. Besides, u' is not within the transmission range of any of the nodes on the

boundary. If the nodes on the boundary of the hole employ restricted flooding to their one-hop neighbouring nodes, the stuck packet does not have to go back to the void node p along the boundary of the hole. Therefore, in the example of Figure 20, node v' will get the flooding packet and then can greedily forward it towards destination q through u' and the other nodes inside the hole. As opposed to the BOUNDHOLE alone, this approach can guarantee delivery to the destination [37].

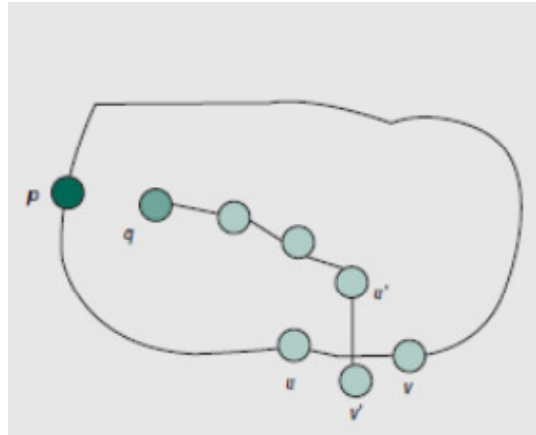


Figure 21. An example for BOUNDHOLE plus Restricted Flooding

Active Exploration plus Passive Participation: This hybrid approach better suits networks where the nodes have adjustable power (and thus transmission range) but the maximum power is limited. Similar to the Active Exploration scheme, if a node encounters a void, it gradually increases its power up to a maximum value. If it cannot still locate a forwarding node during the active exploration phase, the node enters into passive participation, not to forward any subsequent packets for that destination [55].

Coordinate Depth Forwarding plus GPSR-like algorithms: This hybrid approach is proposed for wireless sensor networks [56]. As the first step it proposes a greedy routing technique called Coordinate Depth Forwarding (CDF). Forwarding progress is based on hop count distance from base stations. Each node creates and maintains a base station depth table, which contains base stations' location information. Depth distance information is broadcast by the base stations and then propagated in the network by the nodes. Depth forwarding finds routes only to the base stations. Using its base station table, each node forwards the packet to its nearest base stations. To employ the CDF technique for void handling, the authors propose a hybrid approach. First,

greedy forwarding is performed. Once it fails, the protocol switches to CDF mode before any GPSR-like algorithm is activated to handle the void. The goal is to resolve voids in a localized manner, using base stations as anchors. A base station with the smallest angle to the source-destination line is selected and then the nodes forward the packet using anchor points being the base stations. If the CDF mode fails too, the protocol employs a GPSR-like algorithm (a variation of face routing). More details of this approach can be found in [56].

3. Discussion

There are some shortcomings and problems with the reviewed protocols. More importantly, the majority of the reviewed protocols rely on unrealistic assumptions. Since the target network for our future work is Unmanned Aeronautical Ad hoc Networks (UAANET), the unrealistic assumptions become more pronounced from this perspective considering the special characteristics of such networks. Unmanned Aeronautical Ad hoc Networks are formed as mobile and usually sparse networks in a three dimensional environment (we have done a survey on UAANETs network issues; please see [74]) A routing protocol to be used in UAANETs must be designed with the special characteristics of such networks in mind. In this section, we will evaluate the applicability of the reviewed protocols to UAANETs. For this purpose, we look at different factors such as: performance under mobility, communication overhead, applicability to 3D networks, complexity and performance in case of inaccurate location information.

First, we look at the nature of planar graph based algorithms. Planarization techniques, such as Gabriel Graphs and Relative Neighbourhood Graphs (RNG) (reviewed in Section 2.2) are based on Unit Disk Graphs (UDG). That is, all nodes must have equal transmission ranges. This is not true in real-life and especially for obstructed scenarios. The basic principle of planarization techniques fails when applied to scenarios with obstacles [56]. In other words, applying planarization techniques may disconnect a connected network with particular patterns of obstacles between nodes. In addition, asymmetric radio links and inaccurate node location information could cause errors that violate the assumption of the unit disk graph. The causes and effects of planarization failures are discussed in details in [57]. These failures motivated studying approaches that are not based on planarization such as Cross Link Detection Protocol (CLDP) [58], which was criticized to be complex and costly in [59]. So it led to another proposal, which employs a Greedy Distributed Spanning Tree Routing (GDSTR) approach [59]. It is said that GDSTR is also robust to location errors [59]. Although it is mentioned in GDSTR that it does not require radio ranges to be uniform [59], the protocol is simulated with unit radio ranges. However, this protocol also suffers from a number of limitations that are discussed in this section. For more details, please see [59].

One other unrealistic design factor with planar graph based techniques is that they are studied in a static network at the routing layer, for example, by assuming that routing takes places faster than network changes. Implementation of the Face-2 algorithm is not done for dynamically changing networks [24]. GOAFR+ also executes a routing algorithm on temporarily stationary nodes and leaves the integration of network movement for future work. It is explicitly mentioned in GOAFR+ that it assumes routing takes place much faster than node movement [35]. Overall, it is under such unrealistic assumptions that different face routing protocols claim to guarantee delivery of the packet to the destination [24].

Improvements are made to planarization-based algorithms such as proposals to improve the efficiency of the paths by for example identifying a better path as in ProgressFace of [60] or reducing the number of hops as in VR-forwarding of [61] or the non-message less proposal by [62], which is a technique that exchanges messages to construct a planar graph. However, routing algorithms based on planarization suffer disconnection problems in sparse networks [63].

The most important drawback with planar graph based algorithm from UAANETs' point of view is that they are not applicable to three dimensional environments. Planarization techniques only work in two dimensional coordinate systems as the name implies. Besides, all of the other discussed studies in this report consider voids in a two dimensional coordinate system. There are a number of proposals for 3D georouting for wireless ad hoc and sensor networks. Some of them include [64, 65, 66, 67, 68 and 69]. Studying the 3D georouting protocols in detail is among our future work.

Communication overhead is an important evaluating factor, which refers to the number of messages that are required to be transmitted when handling a void [37]. Face routing has high overhead. In addition, most planar graph based techniques require periodic one-hop beacons. Flooding based protocols also incur a lot of overhead due to the nature of flooding. Among the reviewed protocols, passive participation and active exploration have low overhead.

Some of the reviewed protocols have high complexity in that they are difficult to implement, or require extra resources or have complex processing. Of those with high complexity we can mention Partial Hop-by-hop Routing, Partial Source Routing, Anchored Geodesic Packet Forwarding, alternate network, active exploration and all hybrid void handling techniques.

Flooding, one-hop flooding, passive participation and void avoidance are simple. The rest are of medium complexity [37].

Cost-based protocols have relatively high complexity and overhead to maintain and adjust the cost for optimal paths [56]. In fact, cost-based void handling techniques are designed with a static network in mind. Otherwise, network performance will get worse due to too much overhead imposed by cost adjustment and maintenance.

Inaccurate location information is another issue that is worth considering in real-life scenarios. All of the reviewed protocols assume accurate position information. There are studies about the effect of inaccurate location information on greedy forwarding and how to make them more tolerant of inaccurate position information. For a few of the studies on this subject please refer to [70, 71, and 72].

Overall, for our future research, we are interested in designing a light-weight, low-cost geographic routing protocol that works well in both 2D and 3D coordinate systems and in the presence of node mobility in a 3D space. In other words, topology changes must be taken into account. To the best of our knowledge, there is no clear understanding of the behaviour of discussed recovery strategies when the topology of the network changes due to mobility, while a packet is being forwarded. In addition, the planar based geographic routing protocols claim to guarantee delivery of the packet to the destination in 2D networks [24]. However, as mentioned before, these techniques are studied in a static network. Besides, planar graph based techniques are not applicable to 3D networks. Durocher et al. [75] proved that there does not exist a deterministic local routing algorithm for 3D networks that guarantees delivery of messages. Thus, for any solution for geographic routing in 3D networks, we cannot expect 100% guarantee for delivery, especially while the nodes are moving. Therefore, this is an issue to investigate in our future research as well. Under all the realistic assumptions, we aim for a geographic protocol that works reasonably well in UANNETs. The performance of greedy geographic forwarding in UANNETs is studied in [73], which sets the way for more in-depth research for our future work.

The following section introduces the basic idea of our proposal in-line with the goals mentioned above. The proposed solution employs a random approach as for recovery strategy once greedy forwarding fails. As the future work, we would like to investigate the efficiency and performance

of our proposed protocol by looking at different metrics such as total number of hops per route (i.e., total number of hops each packets takes to its destination), overhead, which could be measured as the total number of routing traffic sent in the entire network, end-to-end delay and packet delivery ratio (i.e., data traffic received / data traffic sent). We plan to collect the statistics by conducting simulations.

4. Randomized Geographic Forwarding (RGF)

Like the geographic routing protocols discussed in Section 2, our proposed solution is based on greedy forwarding as the core of the protocol. Therefore, the preliminaries of a greedy forwarding protocol are in place. Below are the preliminaries and assumptions of the greedy forwarding protocol on top of which we build our protocol:

- All nodes have their location information available.
- An initial flooding is performed throughout the whole network so that each node will acquire the initial position of every other reachable node in the network. This information will be stored in a destination table.
- Flooding is also done based on the distance a node moves, which can be specified by the user. This will result in updating the destination table. However, in the initial phase of our research and for simplicity, we assume there is only one static destination in the network.
- All nodes identify their neighbours and their positions by sending Hello messages. In other words, every node broadcasts a hello message periodically. The information received in Hello messages is to update the neighbour table maintained by each node in the network. Only if the information is newer than the corresponding entry in the table or if it is new (i.e., from a new neighbour) will the entry be updated or created. More specifically, the information that is kept in the neighbour table is as follows: the neighbour's latitude and longitude, timestamp and timeout. Timeout is determined by neighbour expiry time, which can be set by the user. The entry for which the current time is greater than timeout, will be removed from the neighbour table.

Our contribution, however, is to introduce a recovery strategy when greedy forwarding fails. We refer to this strategy as Randomized Geographic Forwarding (RGF). The general idea of this algorithm is as follows: when a node A receives a packet to forward greedily but encounters a local minimum (i.e., node A finds no neighbour closer to the destination than itself), it will switch to RGF as recovery strategy. In RGF, node A selects a neighbour randomly from the set of its current neighbours and forwards the packet to this neighbour. It is very probable that this neighbour will be able to switch back to greedy forwarding. Therefore, by default, the neighbouring node who receives the packet will forward the packet greedily unless it concludes

that it is a void node too. In this case, the neighbouring node will employ RGF to overcome the void situation.

We propose two approaches in RGF, resulting in two versions of RGF. For simplicity, version 1 considers all neighbours of a node equally likely to be selected. That is, a neighbour will be chosen uniformly at random from the set of a node's neighbours. Version 2 assigns different weights to each neighbour relative to its distance from the destination. The neighbour to be selected will be determined by its relative weight. For example, we can assign the weights in such a way that the neighbour with shorter distance to the destination will have higher probability to be selected as the next hop. Therefore, we perform a weighted random sampling on the neighbours. Weight assignment is discussed in the following section. The goal in the design of this algorithm is to have flexibility in weight assignment so that we will be able to study the effect of nodes' distances from the destination in being selected as the next hop in different scenarios.

If node A is to perform RFG and if it has n neighbours, assuming that each neighbour is assigned an appropriate weight, we define the main part of the algorithm as follows:

Let $P(i) = w(i) / \sum_{j=1}^n w(j)$ be the probability of neighbour i to be selected where $1 \leq i \leq n$, then select one.

4.1 Weight Assignment

This section discusses the weight assignment for each of the two versions of RGF protocol.

Version 1. In the first version, as discussed in the previous section, we consider all neighbours of node A to have equal probability to be chosen as the next hop. Assuming node A has n neighbours:

$$w(1) = w(2) = \dots = w(n)$$

and following the formula for probability given above:

$$P(1) = P(2) = \dots = P(n) = 1/n$$

Version 2. In the second version, we assign weights to each node relative to its distance from the destination. In this proposal, we suggest giving higher weights and thus higher probabilities to neighbours that are closer to the destination. This could be justified in that we detour less from the destination. Giving weights relative to the distance from the destination implies that node A , on switching to RFG, must perform a small calculation to assign the weights to its current neighbours. In other words, once node A determines that it must overcome a void by employing RFG, it will have to calculate the distance from the destination for each and every node of its neighbours. The coordinates of the destination exists in the packet that node A has received. Also, node A has already received the coordinates of all its neighbours as this information is updated in node A 's neighbour table upon reception of a Hello message. Assuming a total of n neighbours for node A and defining d_i to be the distance of neighbour i from the destination, the weight for neighbour i would be as follows:

$$w(i) = 1/d_i \quad \text{where } 1 \leq i \leq n$$

and following the formula for calculating the probability proposed in the previous section, we would have:

$$P(i) = (1/d_i) / \sum_{j=1}^n 1/d_j \quad \text{where } 1 \leq i \leq n$$

Once node A calculates the probabilities for each of its neighbours, it will perform a weighted random selection to choose one neighbour as the next hop so that it can recover from the void situation.

4.2 Discussion

A number of issues need to be considered in both versions of this approach; they are listed below and then we will discuss them subsequently.

1. Whether or not a node, performing RGF, needs to exclude the neighbour from whom it received the packet.
2. Are loops likely to happen?

Suppose node *A* forwards the packet to node *B*, at which it faces a void. The first issue can be described as follows. Node *A* is a neighbour to node *B*. Therefore, it is among node *B*'s neighbours in the neighbouring table. Now, employing RGF, node *B* randomly selects one of its neighbours as the next hop. Thus, node *A* could also be selected. However, our suggestion is to exclude node *A* at this stage as it has tried forwarding the packet once. We give the chance to node *B*'s other neighbours, many of which are likely to be closer to the destination than node *A*. . Having said that, RGF will go back to node *A* only when node *B* finds no neighbour other than node *A*. In this case, node *A* must be informed by an *R* flag that it is receiving the packet from a neighbour who faced a void and attempted to switch to RGF but failed. Therefore, the flag is an indicator to node *A* that it must now forward the packet to its second closer neighbour to the destination (other than *B*) if any.

The second issue is related to encountering loops. Loops happen when a packet circles among a certain number of nodes over and over again and thus gets stuck. Figure 22 shows an example when a loop can occur. Starting with greedy forwarding, the source *S* forwards the packet to node *A* and then node *A* forwards it to node *B*. Node *B* encounters a void as it does not have any neighbour closer to destination *D* than itself. Therefore, it switches to RGF. Node *B* has no choice but to choose node *C* as the next hop because node *B*'s other neighbour is node *A* that will be excluded as the packet was received from it. Node *C* starts with greedy forwarding and thus forwards the packet to node *B*, which is closer to the destination. Node *B* again employs RGF and this time has two neighbours to randomly select from. In either case (node *A* or node *C*), the scenario explained above will be repeated. Therefore, we observe a kind of a loop, in which the packet gets stuck.

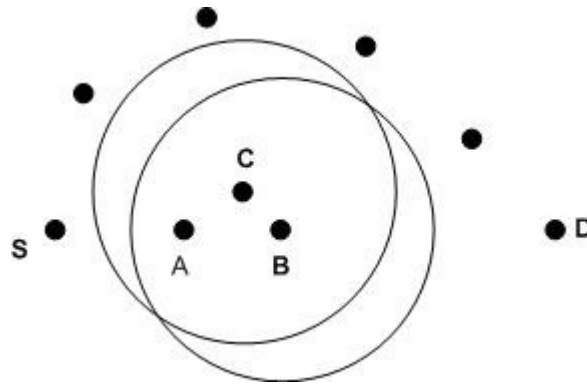


Figure 22. An example of a loop occurring in RGF

At this stage we choose not to react to loop occurrences. Eventually, based on packet's TTL, the loop will stop. In fact, we will lose the packet for the sake of not imposing complexity to the protocol.

5. Conclusion

There are many geographic routing protocols proposed for Mobile Ad Hoc Networks. They are based on greedy forwarding, in which the packet is forwarded to a neighbour who is closer to the destination. The problem, however, is that the packet encounters a void if the packet holder does not have any neighbour closer to the destination than itself. Therefore, the discussed protocols in this report each propose a recovery strategy to recover from void situations. The most prominent solution is called face routing, which is based on planar graphs. A broad overview of face routing protocols is given in this report and the problems with planar graph based algorithms are discussed. They are not applicable to 3D networks. They have high communication overhead and the majority of these protocols are based on unrealistic assumptions. For example, almost all of them are evaluated in a static network rather than a mobile network. We have also looked at solutions other than face routing as recovery strategy. At the end, we have proposed our solution as recovery strategy considering realistic features of a MANET. Our proposal is called Randomized Geographic Forwarding (RGF) and is based on random selection among a node's neighbours when it faces a void. We consider two approaches, resulting in two versions of RGF. One approach is to consider all neighbours equally likely to be chosen and the other approach is to assign weights to them so that they will have different probabilities to be chosen. The details of RGF are discussed in this report. The performance evaluation of RGF is a topic to be studied in the future.

References

- [1] J. Blum, A. Eskandarian, and L. Ho_man, “Challenges of intervehicle ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 347 – 351, Dec. 2004.
- [2] E. Sakhaee, A. Jamalipour, and N. Kato, “Aeronautical ad hoc networks,” in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, April 2006.
- [3] M. Mauve, J. Widmer and H. Hartenstein, A Survey on Position Based Routing in Mobile Ad-hoc Networks, *IEEE Network Magazine*, 15(6):30–39, November 2001.
- [4] C. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance vector Routing (dsv) for Mobile Computers,” *Comp. Commun. Rev.*, Oct. 1994, pp. 234–44.
- [5] P. Jacquet *et al.*, “Optimized Link State Routing Protocol. Internet draft, draft-ietf-manet-olsr-04.txt, work in progress, Sept. 2001.
- [6] C. Perkins and E. Royer, “Ad-hoc on-demand Distance Vector Routing,” *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. App.*, Feb. 1999, pp. 90–100.
- [7] D. Johnson and D. Maltz, *Mobile Computing*, Chap. 5 — Dynamic Source Routing, Kluwer Academic Publishers, 1996, pp. 153–81.
- [8] Z. Haas and M. Pearlman, “The Performance of Query Control Schemes for the Zone Routing Protocol,” *ACM/IEEE Trans. Net.*, vol. 9, no. 4, Aug. 2001, pp. 427–38.
- [9] E. Royer and C.-K. Toh, “A Review of Current Routing Protocols for Ad Hoc Wireless Networks,” *IEEE Pers. Commun.*, Apr. 1999, pp. 46–55.
- [10] J. Broch *et al.*, “A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols,” *Proc. 4th ACM/IEEE Int’l. Conf. Mobile Computing and Networking MOBICOM ’98*, Dallas, TX, USA, 1998, pp. 85–97.
- [11] E. Kaplan, *Understanding GPS*, Artech House, 1996.
- [12] S. Capkun, M. Hamdi, and J. Hubaux, “Gps-free Positioning in Mobile Ad Hoc Networks,” *Proc. Hawaii Int’l. Conf. System Sciences*, Jan. 2001.
- [13] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57–66.
- [14] Ko, Y., and Vaiyda, N. Location-aided routing in mobile ad hoe networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, Dallas, Texas, USA, Aug. 1998.
- [15] Finn. *Routing and Addressing Problems in Large Metropolitan-scale Internetworks*. Technical Report ISI/RR-87-180, USC/ISI, March 1987.

- [16] Hou and V.O.K. Li. Transmission Range Control in Multihop Packet Radio Networks. *IEEE Transactions on Communications*, 34(1):38-44, 1986.
- [17] Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3):246-257, 1984.
- [18] T.-C. Hou and V. O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Commun.*, vol. 34, no. 1, Jan. 1986, pp. 38–44.
- [19] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. 11th Canadian Conf. Comp. Geo.*, Vancouver, Aug. 1999.
- [20] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Commun.*, vol. 32, no. 3, Mar. 1984, pp. 246–57.
- [21] T.-C. Hou and V. O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Commun.*, vol. 34, no. 1, Jan. 1986, pp. 38–44.
- [22] I. Stojmenovic and X. tin, "bop-Free Hybrid Single~poth/Flooding Routing Algorithms with Guoronteed Delivery for Wireless Networks," *IEEE Tronr. Pomllet and Dirtrib. Sys.*, vol. 12, Oct. 2001, pp. 1023-32.
- [23] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conf. Computational Geometry*, Vancouver, BC, Canada, Aug. 1999, pp. 51–54 [Online]. Available: citeseer.nj.nec.com/kranakis99compass.html
- [24] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006.
- [25] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.
- [25] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.
- [27] 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.
- [28] 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.

- [29] 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.
- [30] H. Frey. “Scalable Geographic Routing Algorithms for Wireless Ad Hoc Networks.” *IEEE Network*, Vol. 18, Jul./Aug. 2004, pp. 18-22.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proc. PODC 2003*, Boston, MA, USA, July 2003.
- [36] 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.
- [37] Chen, D., Varshney, P.K.: A survey of void handling techniques for geographic routing in wireless networks. *Communications Surveys and Tutorials*, 50–67 (2007)
- [38] M. Heissenbüttel et al., “BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks,” *Elsevier Computer Commun. J.*, vol. 27, no. 11, July 2004.
- [39] Y. Xu et al., “PSGR: Priority-based Stateless Geo-Routing in Wireless Sensor Networks,” *Proc. IEEE MASS 2005*, Washington D.C., November, 2005.
- [40] Q. Fang, J. Gao, and L. J. Guibas, “Locating and Bypassing Routing Holes in Sensor Networks,” *Proc. IEEE INFOCOM 2004*, Hong Kong, Mar. 2004.
- [41] R. Jain, A. Puri, and R. Sengupta, “Geographical Routing Using Partial Information for Wireless Ad Hoc Networks,” *IEEE Pers. Commun.*, Feb. 2001, pp. 48–57.

- [42] I. Stojmenovic, M. Russell, and B. Vukojevic, "Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks," *Computers and Informatics*, vol. 21, no. 2, 2002, pp. 149–65.
- [43] D. Chen and P. K. Varshney, "On Demand Geographic Forwarding for Data Delivery in Wireless Sensor Networks," *Elsevier Computer Commun. J.*, special issue on Network Coverage and Routing Schemes for Wireless Sensor Networks, 2006.
- [44] F. Ye, A. Chen, S. Lu, L. Zhang, "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks," *Proc. IEEE ICCCN 2001*, 2001.
- [45] L. Zou, M. Lu, and Z. Xiong, "PAGER-M: A Novel Locationbased Routing Protocol for Mobile Sensor Networks," *Proc. Broadwise 2004*, San Jose, CA, Oct. 2004.
- [46] S. Chen, G. Fan, and J. Cui, "Avoid 'Void' in Geographic Routing for Data Aggregation in Sensor Networks," *Int'l. J. Ad Hoc and Ubiquitous Computing (IJAHUC)*, special issue on Wireless Sensor Networks, vol. 2, no. 1, 2006.
- [47] 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, Oct. 2001.
- [48] D. Chen, J. Deng, and P. K. Varshney, "On the Forwarding Area of Contention-Based Geographic Forwarding for Ad Hoc and Sensor Networks," *Proc. IEEE SECON 2005*, Santa Clara, CA, Sept. 2005.
- [49] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, Oct.–Dec. 2003.
- [50] D. Chen, J. Deng, and P. K. Varshney, "A State-Free Data Delivery Protocol for Wireless Sensor Networks," *Proc. IEEE WCNC 2005*, New Orleans, LA, Mar. 2005.
- [51] T. He et al., "SPEED: A Stateless Protocol for Real-time Communication in Sensor Networks," *Proc. Int'l. Conf. Distributed Computing Systems*, Providence, RI, May 2003.
- [52] D. S. J. De Couto and R. Morris, "Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding," Technical report MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.
- [53] L. Blazevic, S. Giordano, and J. Y. Le Boudec, "Self Organized Terminode Routing," *J. Cluster Computing*, vol. 5, no. 2, Apr. 2002.
- [54] D. Chen and P. K. Varshney, "On Demand Geographic Forwarding for Data Delivery in Wireless Sensor Networks," *Elsevier Computer Commun. J.*, special issue on Network Coverage and Routing Schemes for Wireless Sensor Networks, 2006.

- [55] D. Chen, J. Deng, and P. K. Varshney, "On the Forwarding Area of Contention-Based Geographic Forwarding for Ad Hoc and Sensor Networks," Proc. of IEEE SECON, Santa Clara, California, September, 2005.
- [56] Gustavo Weber Denardin, Carlos Henrique Barriquello, Alexandre Campos, and Ricardo Nederson do Prado. "A geographic routing hybrid approach for void resolution in wireless sensor networks", *J. Syst. Softw.* 84, 10, October 2011.
- [57] 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.
- [58] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), May 2005.
- [59] Leong, B., Liskov, B., & Morris, R. (2006). Geographic routing without planarization. In 3rd Symposium on Networked Systems Design & Implementation (NSDI), San Jose, California, USA, May 2006.
- [60] C.H. Lin, S.A. Yuan, S.W. Chiu, and M.J. Tsai, "Progressface: An algorithm to improve routing efficiency of GPSR-like routing protocols in wireless ad hoc networks", "IEEE Transactions on Computer", vol. 59, no. 6, pp. 822 – 834, June 2010.
- [61] Young Il Ko, Chang-Sup Park, In Chul Song, and Myoung Ho Kim. 2009. An efficient void resolution method for geographic routing in wireless sensor networks. *J. Syst. Softw.* 82, 963-973, June 2009.
- [62] H. Frey and I. Stojmenovic. "On Delivery Guarantees and Worst Case Forwarding Bounds of Elementary Face Routing Components in Ad Hoc and Sensor Networks". IEEE Trans. on Computers, 2011.
- [63] SEADA, K., HELMY, A., AND GOVINDAN, R. Localization errors on geographic face routing in sensor networks. In Proc. IEEE IPSN Workshop, Berkeley, CA, USA, Apr. 2004.
- [64] A.E. Abdallah, T. Fevens, J. Opatrny, and I. Stojmenovic, "PowerAware Semi-Beaconless 3D Georouting Algorithm Using Adjustable Transmission Ranges for Wireless Ad Hoc and Sensor Networks," *Ad Hoc Networks*, vol. 8, no. 1, pp. 15-29, 2010
- [65] S. S. Lam and C. Qian. Geographic Routing in d-dimensional Spaces with Guaranteed Delivery and Low Stretch. In Proceedings of ACM SIGMETRICS, June 2011.
- [66] J. Zhou, Y. Chen, B. Leong, and P. Sundar, "Practical 3D Geographic Routing for Wireless Sensor Networks," in Proc. of SenSys, pp. 337–350, 2010.

- [67] Zhixiao Wang; Deyun Zhang; Alfandi, O.; Hogrefe, D.; , "Efficient geographical 3D routing for Wireless Sensor Networks in smart spaces," Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future , vol., no., pp.168-172, 16-18 Feb. 2011.
- [68] R. Flury and R. Wattenhofer. Randomized 3D geographic routing. In Proceedings of the IEEE INFOCOMM 2008, 2008.
- [69] Manas Kumar Mishra and M. M. Gore. "An improved forwarder selection approach for energy aware geographic routing in three dimensional wireless sensor networks", In *Proceedings of the 2011 International Conference on Communication, Computing & Security (ICCCS '11)*, ACM, New York, NY, USA, 166-171, 2011.
- [70] Y. Kim, J. Lee, and A. Helmy, "Impact of location inconsistencies on geographic routing in wireless networks," in Proc. ACM MSWIM'03, pp. 124–127, 2003.
- [71] D.Son, A.Helmy, B.Krishnamachari. "The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad Hoc and Sensor Networks: Analysis and Improvement Using Mobility Prediction", in IEEE Transactions on Mobile Computing, vol. 3, pp. 233-245, July 2004.
- [72] H. Frey, X. Li, S. Ruehrup, R. Marin-Perez, P. Ruiz, "Effective Geographic Routing in Wireless Sensor Networks with Inaccurate Location Information", Ad-hoc, Mobile, and Wireless Networks, Lecture Notes in Computer Science- Springer Berlin / Heidelberg, 2011.
- [73] Rostam Shirani, Marc St-Hilaire, Thomas Kunz, Yifeng Zhou, Jun Li, and Louise Lamont, "The performance of greedy geographic forwarding in unmanned aeronautical ad-hoc networks", *Proceedings of the 9th Conference on Communications Networks and Services Research (CNSR 2011)*, Ottawa, Canada, pages 161-166, May 2011.
- [74] R. Shirani, "Reactive-greedy-reactiveinunmannedaeronautical ad-hoc networks: Acombinational routingmechanism," Master's thesis, CarletonUniversity, 2011. [Online]. Available: <http://www.csit.carleton.ca/~msthilaire/Thesis/Rostam%20Shirani.pdf>
- [75] S. Durocher, D. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. *Wireless Networks*, 16(1):227–235, 2010