Select-and-Protest-based Beaconless Georouting with Guaranteed Delivery in WSNs

Hanna Kalosha, Amiya Nayak, Stefan Rührup, Ivan Stojmenović

School of Information Technology & Engineering
University of Ottawa, 800 King Edward Avenue
Ottawa, Ontario K1N 6N5, Canada

Abstract Recently proposed beaconless georouting algorithms are fully reactive, with nodes forwarding packets without prior knowledge of their neighbors. However, existing approaches for recovery from local minima can either not guarantee delivery or they require the exchange of complete neighborhood information.

We describe two general methods that enable completely reactive face routing with guaranteed delivery. The Beaconless Forwarder Planarization (BFP) scheme finds correct edges of a local planar subgraph at the forwarder node without hearing from all neighbors. Face routing then continues properly. Angular Relaying determines directly the next hop of a face traversal. Both schemes are based on the Select and Protest principle. Neighbors respond according to a delay function, but only if there is no other neighbor within their forbidden region. Protest messages are used to correct occasionally wrong selections by neighbors that are not in the planar subgraph.

We show that a correct beaconless planar subgraph construction is not possible without protests. We also show the impact of the chosen planar subgraph construction on the message complexity. This leads to the definition of the Circlunar Neighborhood Graph (CNG), a new proximity graph, that enables BFP with a bounded number of messages in the worst case, which is not possible when using the Gabriel graph (GG). The CNG is sparser than the GG, but this does not lead to a performance degradation. Simulation results show similar message complexities in the average case when using CNG and GG.

Angular Relaying uses a delay function that is based on the angular distance to the previous hop. We develop a theoretical framework for delay functions and show both theoretically and in simulations that with a function of angle and distance we can reduce the number of protests by a factor of 2 in comparison to a simple angle-based delay function.

1 Introduction

Wireless ad hoc and sensor networks consist of nodes which are equipped with very limited resources. They usually have a wireless transceiver with limited transmission range, restricted memory and processing capabilities, and, above all, limited energy resources. Many effort has been invested in developing topology control strategies and routing protocols in order to increase the energy-efficiency. In this paper we focus on beaconless geographic routing, a completely reactive approach for multi-hop communication in wireless ad hoc and sensor networks, that relies on position information and reduces the overhead for the exchange of topology and routing information to a minimum.

Beaconless georouting algorithms work completely reactive and reduce the overhead for exchanging topology and routing information to a minimum. They follow the principle of geographic routing, where a message is routed to the location of the destination instead of a network address. This is based on the assumptions that each node can determine its own geographic position and that the source knows the position of the destination. The use of position data enables routing without routing tables or prior route discovery. Conventional geographic routing algorithms use two basic forwarding principles: *greedy forwarding* and *face routing*. Greedy forwarding means to select a neighbor that minimizes the distance to the target. This strategy fails in case of a local minimum, i.e. if no neighbor is closer to the destination. Then, face routing can be used in order to recover from this situation. The message is routed along the incident face of the communication graph using the right-hand rule until a position is found that is closer to the destination than the local minimum. Face traversals work only on a planar subgraph, otherwise crossing edges might cause a routing loop. Thus, a local *planarization* strategy is needed, which determines the edges of a planar subgraph.

Beaconless Routing

Conventional geographic routing algorithms rely on the position information of their 1-hop-neighbors. This information can be gathered by a periodic exchange of *beacon* messages. Beaconless routing algorithms try to avoid this message exchange and provide a completely reactive routing. The basic principle of beaconless forwarding is the following: The *forwarder*, i.e. the node that currently holds the packet, broadcasts it to its neighbors. The nodes within the forwarder's transmission range receive the packet, but only the nodes in the *forwarding area* are eligible for forwarding it further (see Fig. 1). These nodes are called *candidates*. The most suitable candidate is determined by a contention mechanism: After receiving the packet, each candidate starts a timer. The timer is determined by a *delay function* that favors the most promising node, e.g., the node closest to the destination has the shortest timeout. This node forwards the packet again, when its timer expires. The other candidates notice that the packet is re-transmitted and cancel their timers. This strategy follows the greedy principle, because it uses always locally optimal decisions.

The Beaconless Recovery Problem

As greedy routing fails in case of a local minimum, a recovery strategy is needed to guarantee delivery. The preferred recovery method for conventional geographic routing is the face traversal on a planar subgraph, which is constructed from neighborhood information. But in beaconless routing the full knowledge of the neighborhood is not a priori available. Instead, part of this knowledge has to be gained by exchanging messages, if it is not implicitly given by the location of the nodes. Therefore, we can describe the beaconless recovery problem by two questions, whose answer is the key to guaranteed delivery:

1. How to construct a local planar subgraph on the fly?
2. How to determine the next edge of a planar subgraph traversal?

The beaconless recovery problem has to be solved reactively and with as few messages as possible. Existing approaches use a reactive message exchange in which all neighbors are involved in the worst case. This rises the question, whether we can reduce this message overhead and thus achieve a significant message reduction in comparison to conventional protocols that rely on beaconing.

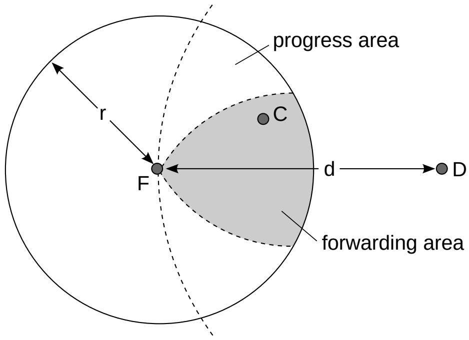


Figure 1: *Forwarder (F), candidate (C) and destination (D)*

In this paper we answer this question and provide solutions for both variants of the beaconless recovery problem: Beaconless Forwarder Planarization (BFP) first constructs an approximation of the planar subgraph and then sorts out nodes that are not neighbors in a planar subgraph. We use proximity graphs such as Gabriel graph and relative neighborhood graph for the planar subgraph construction because edges in these graphs can be determined locally. We propose the Circlunar Neighborhood Graph (CNG), a planar proximity graph that can be constructed with less messages than the Gabriel graph and that has a better connectivity than the relative neighborhood graph. The second solution of the beaconless recovery problem is Angular Relaying, which first tries to find the next neighbor of a right-hand face traversal and then switches to another neighbor, if the selected neighbor is not adjacent in the planar Gabriel subgraph.

Overview

In Section 2 we review related work. Section 3 describes the Beaconless Forwarder Planarization method which provides the general framework of creating planar subgraphs reactively for face routing. In Section 4 we determine the crucial properties of planar subgraph constructions for the efficiency of BFP. In Section 5 we introduce the Circlunar Neighborhood Graph, a new proximity graph which has advantageous properties for local subgraph construction in beaconless protocols. It reduces the message overhead to a constant number while providing better connectivity than the relative neighborhood graph. Section 6 describes the Angular Relaying method, an alternative solution to the beaconless recovery problem. In Section 7 we present simulation results for the aforementioned protocols.

2 Related Work

One building block of geographic routing strategies are greedy forwarding strategies. They are based on position-based progress criterions such as MFR [TK84] or the *greedy method* [Fin87]. Progress in terms of MFR means to decrease the distance of the projection on the straight line to the target, while the greedy method simply refers to the Euclidean distance. The first beaconless routing algorithms, BLR [HB03], CBF [FWMH03], and IGF [BHSS03], use these greedy criterions to define the delay functions, which determine the candidate with the most progress by giving him the shortest timeout. There are further protocols addressing specific problems of the initial approaches. Blind Geographic Routing (BGR) [WT05] contains a strategy to avoid simultaneous transmissions. Geographic Random Forwarding (GeRaF) [Zor04] divides the forwarding area into zones and select the next forwarder by contention among the nodes within these zones. All these approaches work well in dense networks, where there is always a neighbor closer to the destination. If this is not the case and the greedy algorithm faces a local minimum, delivery can only be guaranteed, if a recovery from that situation is possible. Recovery strategies have been developed for geographic routing algorithms (see [CV07] for a survey) and many of them are based on face traversals using a planar subgraph. Prominent subgraph constructions are the Gabriel graph (GG) [GS69] and the relative neighborhood graph (RNG) [JT92], but also localized variants of the Delaunay triangulation have been proposed [GGH$^+$01, LCW02, LSW04]. Face routing on a planar subgraph in combination with

Protocol	Empty Forwarding Area	Recovery (from local minima)	Guaranteed delivery
BLR	use MFR area	Beaconing + face routing	yes
CBF	use progress area	(left open)	??
IGF	–	–	no
BGR	rotate fwd. area	–	no
GeRaF	– *	–	no
PSGR	– *	Bypass	??
NB-FACE	– **	Clockwise timeout and Gabriel neighbor selection	yes
GDBF	– **	Distance-based timeout, Gabriel neighbor selection	yes

*) Forwarding area covers the complete progress area

**) Forwarding area covers the complete transmission area

Table 1: *Beaconless routing protocols and their recovery methods*

greedy forwarding is the idea behind the Greedy-Face-Greedy algorithm (GFG) [BMSU99], which became a standard technique for geographic routing.

2.1 Beaconless Recovery

While the recovery problem is well studied for geographic routing algorithms, the beaconless approaches leave room for improvement. In beaconless routing, the term "recovery" is often used in connection with heuristics, that enlarge the set of possible candidates, if the forwarding area is empty, but do not guarantee delivery. BLR, CBF and BGR use this kind of heuristic. PSGR [XLXM05] contains a more sophisticated recovery mechanism, however the delivery is questionable, as no crossing-free subgraph is considered.

The following beaconless protocols contain a "real" recovery strategy and can thus give delivery guarantees (cf. Table 1). However, all these strategies require position information of the complete neighborhood to be exchanged in the worst case.

BLR Backup mode [HBBW04] (also called *Request-response* approach in [HB03]): The forwarder broadcasts a request and *all* neighboring nodes respond. If a node is closer to the destination, it becomes the next hop. Otherwise the forwarder constructs a local planar subgraph (GG) from the position information of the neighbors and forwards the packet using the right-hand rule. The position when entering backup mode is stored in the packet. Greedy forwarding is resumed when a node is closer to the destination.

Request-Response can be regarded as reactive beaconing, because all neighbors are involved in exchanging position information. The following protocols use an approach, that we classify as *Select and Protest*: they determine possible neighbors of a planar subgraph by a contention process and allow protests afterwards to correct wrong decisions.

NB-FACE [NOH06] is a beaconless variant of the face routing algorithm. The delay function depends on the angle between candidate, forwarder and previous hop such that the first candidate in (counter-)clockwise order responds first. If this node is not a neighbor in the Gabriel graph, then other nodes may protest. The NB-FACE algorithm is similar to a variant of our Angular Relaying scheme (Section 6). However, we will see that NB-FACE yields not always optimal results.

GDBF [CGK$^+$06a, CGK$^+$06b] provides a beaconless Gabriel graph construction and serves as basis for face routing algorithms such as GFG. The local Gabriel subgraph is constructed in two phases, using a timer-based contention mechanism: First, the candidates answer with a delay proportional to their distance to the forwarder, but only if no other neighbor located within their Gabriel circle has responded earlier. The thus constructed subgraph contains directed (asymmetric) edges and is not necessarily planar. Therefore, after the face routing algorithm has selected a candidate that violates the Gabriel graph condition, further nodes may protest against the decision in a second phase. We will see that in the worst case all neighbors have to respond when using the Gabriel graph. GDBF is a variant of our more general BFP scheme.

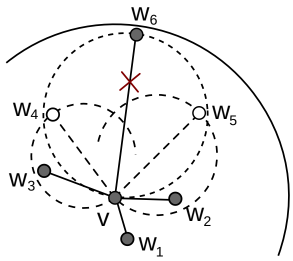


Figure 2: *BFP: Nodes respond in the order w_1, w_2, w_3, w_6 according to their distance to the forwarder v; w_4 and w_5 are hidden. w_4 protests against w_6.*

3 Beaconless Forwarder Planarization

The basic problem of beaconless protocols is that they cannot rely on 1-hop-knowledge. But this knowledge is necessary to build a planar subgraph. Thus, in a recovery situation, the forwarder has to gather information and this is connected with the exchange of messages. In contrast to the Request-Response approach of BLR [HB03], where all neighbors announce their positions upon request, we follow the idea of GDBF [CGK$^+$06b] to reduce the message overhead.

Beaconless Forwarder Planarization (BFP) is a general scheme, that can be used to construct different proximity graphs, such as Gabriel graph and RNG. The BFP algorithm is described in the following. It's message complexity depends on the chosen subgraph. We will later discuss appropriate subgraph constructions and analyze the message complexity.

3.1 The BFP Algorithm

The BFP algorithm consists of two phases, the selection and the protest phase. $N(u,v)$ denotes the proximity region of the chosen subgraph, e.g. the Gabriel circle or the RNG lune over (u,v) (cf. Fig. 8).

1. Selection Phase The forwarder v broadcasts an RTS (including its own position) and sets its timer to $t_{\max}$. Each candidate w sets its contention timer, using the following delay function:

$$t(d) = \frac{d}{r} \cdot t_{\max} \tag{1}$$

(d = distance to forwarder = $|vw|$, r = transmission radius, $t_{\max}$ = maximum timeout). When the contention timer expires, a candidate answers with a CTS. If a candidate w receives the CTS of another node w' that lies in the proximity region $N(u,w)$, then w cancels its timer and remains quiet. We call this mechanism *suppression* and the candidate being suppressed a *hidden node*. Hidden nodes listen to other nodes after their timer expired. If a hidden node w receives the CTS of another node w' with $w \in N(u,w')$, then w' violates the proximity condition and w adds w' to the set of violating nodes S. We call (u,w') a *violating edge*. See also Fig. 2.

2. Protest phase In the second phase, the hidden nodes protest against violating edges. If the set of violating nodes S is not empty, the hidden node w starts its timer, using the same delay function as in the first phase (closest candidates protest first). If w overhears a protest from another hidden node w', then the set of violating nodes has to be checked: A node x can be removed from S, if $w' \in N(u,x)$. When the timer expires and S is not empty, w sends the protest message. The forwarder removes violating edges when it receives protests and finally obtains a planar subgraph.

Task of the forwarder v

1. send the RTS
2. set timer $T = t_{\max}$
   while T is not expired
   upon reception of a CTS from node w:
   add w to the set of neighbors $\Gamma(v)$
3. set timer $T = t_{\max}$
   while T is not expired
   upon reception of a Protest from node w:
   for each node $x \in \Gamma(v)$ do
   if $x \in N(v,w)$ then remove x from $\Gamma(v)$

Task of a candidate w
(after receiving an RTS from forwarder v)

1. hidden = false
2. set timers $T_1 = t(|uw|)$, $T_2 = t_{\max}$
   while T_2 is not expired
   while T_1 is not expired
   upon reception of a CTS from node x
   if $x \in N(v,w)$ then hidden = true
   if (hidden = false) then send CTS and halt
   upon reception of a CTS from node x
   if $w \in N(v,x)$ then
   add x to the set of violating nodes S
3. if $S \neq \emptyset$ then set timer $T = t(|uw|)$
   while T is not expired
   upon reception of a protest from node x
   for each node $y \in S$ do
   if $x \in N(v,y)$ then remove y from S
4. if $S \neq \emptyset$ then send a protest (incl. the own position)

Figure 3: *Beaconless Forwarder Planarization*

4 Proximity Graphs and Beaconless Subgraph Construction

The BFP algorithm can be based on different proximity graph constructions, in order to obtain a planar communication graph (here, it means that the graph is a planar embedding). Most prominent subgraph constructions are Gabriel graph and RNG (cf. [Cim92]):

Definition 1 *The* Gabriel graph *(GG) of a node set V contains an edge (u,v), iff $|uv|^2 \leq |uw|^2 + |vw|^2$ for all $w \in V, w \neq u, v$.*

Definition 2 *The* relative neighborhood graph *(RNG) of a node set V contains an edge (u,v), iff $|uv| \leq \max\{|uw|,|vw|\}$ for all $w \in V, w \neq u, v$.*

The definition implies that two nodes u and v are adjacent, if the so-called proximity region over (u,v) is empty (*proximity condition*). We denote the proximity region with $N(u,v)$. In case of the Gabriel graph, the $N_{GG}(u,v)$ is a circle having $\overline{uv}$ as diameter, in case of the RNG, $N_{RNG}(u,v)$ is a lune over $\overline{uv}$ (cf. Fig. 8). In this paper we assume that all distances are different in order to avoid degenerated cases. However, equal distances can be handled by using $|uv| = (||u-v||_2, key(u), key(v))$ as distance measure [LSW04], where $key(\cdot)$ is based on the node ID or on a lexicographic order of the geographic coordinates. In a similar way, a modified RNG with a constant maximum node degree can be obtained that is still connected on degenerated node sets [Li03].

The choice of the subgraph determines the message efficiency of the BFP algorithm. In the following we will identify the crucial properties to construct a planar and connected subgraph with as few messages as possible.

4.1 Basic Requirements

We consider only undirected, planar, and connected proximity graphs. The proximity region of these graphs is symmetric, it contains at least the Gabriel circle, and it is not larger than the RNG lune.

Lemma 1 *The RNG lune is the maximum proximity region to preserve connectivity.*

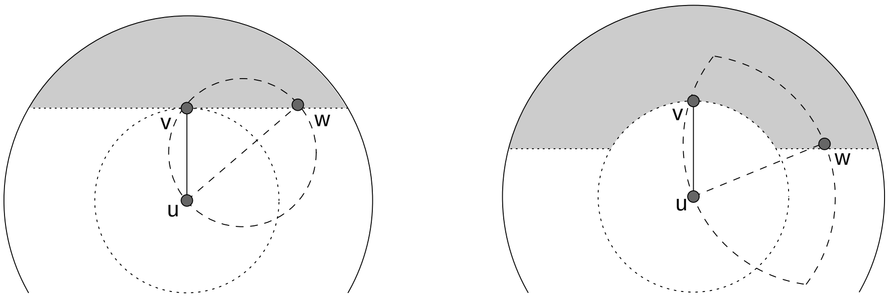


Figure 4: *Suppression region for GG (left) and RNG (right): A node w in the shaded area is not a valid neighbor of u, because v would be inside the Gabriel circle or the RNG lune.*

Proof: Let u,v,w be nodes of an undirected proximity graph, and let $L(u,v)$ denote the RNG lune over (u,v), i.e. the intersection of two circles with radius $|uv|$ centered at u and v. Suppose the proximity region of (u,v) is larger than $L(u,v)$. Then there is a point w outside $L(u,v)$ (i.e. $|uw| > |uv|$ or $|vw| > |uv|$) that belongs to the proximity region and thus invalidates the edge (u,v). If $|uw| < |vw|$ then $u \in L(v,w)$, which disconnects v. Otherwise, $v \in L(u,w)$, which disconnects u. ■

Lemma 2 *The Gabriel circle is the minimum proximity region to obtain planarity.*

Proof: Let $C(u,v)$ denote the Gabriel circle over (u,v), i.e. the circle having $\overline{uv}$ as diameter with its interior. Let m be the midpoint of (u,v). Suppose the proximity region is smaller than $C(u,v)$. Then there is a node w inside $C(u,v)$ with $|mw| < |mu|$, while (u,v) is a valid edge. As G is undirected, the proximity region is symmetric; and this implies that there is another point w' which can be constructed by rotating w by 180° around the midpoint m. Then the circle $C(w,w')$ is inside $C(u,v)$ and empty (because of $|mw| = |mw| < |mu| = |mv|$). Therefore, (w,w') is a valid edge, and it intersects (u,v) in the midpoint, which is a contradiction.

The graph is planar, if the proximity region contains $C(u,v)$: If $C(u,v)$ is empty, then the empty circle rule of the Delaunay Triangulation is also fulfilled for any three nodes. Thus, G is a subgraph of the Delaunay Triangulation, which is planar. ■

4.2 Hidden Nodes and Suppression

The construction of Gabriel graph or RNG is based on the proximity region, which is an empty circle or an empty lune. BFP makes use of this fact to reduce messages: Candidate nodes are suppressed, i.e. they remain quiet, if they would violate this condition.

Definition 3 *The* suppression region *of a node v with respect to u contains all points w with $v \in N(u,w)$, where $N(u,v)$ denotes the proximity region of an edge (u,v).*

Fig. 4 shows the suppression region for Gabriel graph and RNG. In case of the Gabriel graph, w is suppressed, if $\angle uvw < 90°$, and this implies that the border of the suppression region is orthogonal on (u,v). In case of the RNG, $|vw| < |uw|$, and this means that the perpendicular bisector of (u,v) marks the border of the suppression region.

4.3 Ordered Neighborhoods and Protest Messages

In beaconless protocols, the location of the neighbors are not known in advance, but they are revealed one by one when they reply to the forwarder's request. From a graph theoretic point of view, the candidate nodes are inserted into the set of neighbors, and the insertion order is given by the delay function. This determines the resulting neighborhood, because after one node responds, others may be suppressed and remain quiet. In order to formalize this mechanism, we introduce the definition of an *ordered neighborhood*.

Let G denote a graph and $\Gamma(u)$ the set of neighbors of a node u in G. For a node u, define a total order π_u so that $\pi_u(v)$ is the rank of $v \in \Gamma(u)$.

Definition 4 *A node $v \in \Gamma(u)$ is* hidden*, if it is suppressed by a non-hidden node w with smaller rank, i.e. $w \in \Gamma(u)$ with $\pi_u(w) < \pi_u(v)$ and $w \in N(u,v)$.*

Definition 5 *The* π-ordered neighborhood $\Gamma_\pi(u)$ *contains all nodes v for which there is no non-hidden node w with $w \in N(u,v)$.*

An ordered neighborhood can be constructed by inserting nodes one by one, if they fulfill the proximity condition (e.g. empty Gabriel circle). In contrast to the original proximity graph, this condition is only checked for the nodes which have been already added to the neighborhood. Note that in contrast to ordered θ-graphs [BGM04], π defines a local order for each node.

In BFP a distance-based delay function is used (equation 1) which defines the insertion order and determines the neighborhood. The result of Phase 1 of the BFP algorithm is a distance-ordered neighborhood, which contains at least the edges of the desired subgraph.

Theorem 1 *In a proximity graph, the ordered neighborhood of a node v is a superset of the original neighborhood, i.e. $\Gamma_\pi(v) \supseteq \Gamma(v)$.*

Proof: Let v be a neighbor of v, i.e. $u \in \Gamma(v)$. Then, the proximity region $N(v,u)$ is empty and remains empty, regardless of the rank of u. Thus, $u \in \Gamma_\pi(v)$. ■

When constructing the ordered neighborhood, we can be sure, that the nodes of the desired subgraph are included, but there may be violating edges depending on the insertion order. Therefore, Phase 2 of the BFP algorithm is required, where the hidden nodes send protest messages to indicate edges violating the proximity condition. In the worst case, there is one protest message required for each violating edge.

4.4 Distance-ordered neighborhoods

The worst case number of violating edges depends on the order (i.e. the delay function) and also on the chosen subgraph construction. In case of the Gabriel graph, this number is unbounded, whereas it is constant in case of the RNG.

Theorem 2 *A distance-ordered Gabriel neighborhood contains an unbounded number of violating edges.*

Proof: The construction in Figure 5 shows that a node can have $\Theta(n)$ neighbors in its distance-ordered Gabriel neighborhood while it has only one valid Gabriel neighbor. Nodes $w_1, \ldots, w_5$ are placed around v with increasing distance and partially overlapping Gabriel circles as shown in the figure. In the Gabriel neighborhood w_1 inhibits an edge (v,w_2), w_2 inhibits an edge (v,w_3) etc., so that v has only one valid edge. In the distance-ordered neighborhood w_1 is inserted first and w_2 is hidden, because node w_1 is in its Gabriel circle. Node w_3 becomes a neighbor, because w_2 is hidden and not part of the neighbor set. Every second node in the chain will become a neighbor of v, i.e. $\Gamma_\pi(v)$ has a size of $\lceil (n-1)/2 \rceil$. ■

Corollary 1 *The beaconless Gabriel graph construction with a distance-based delay function requires an unbounded number of protests in the worst case.*

The crucial property to bound the number of protests is that a circular sector has to be part of the proximity region (see Figure 6).

Theorem 3 *A distance-ordered neighborhood has at most $\lfloor 4\pi/\theta \rfloor - 1$ violating edges, if the proximity region contains a circular sector of angle θ.*

Proof: Let $\sphericalangle_\theta(u,v)$ be a sector of the circle $C(u,|uv|)$ with angle θ and $\overline{uv}$ as bisecting line (see Fig. 6), and assume that it is contained in the proximity region. A node w is only included in the neighbor set of u, if $\angle vuw > \theta/2$, because of the following reason: If $|uw| < |uv|$, w must be outside $\sphericalangle_\theta(u,v)$; otherwise, v must be outside $\sphericalangle_\theta(u,w)$. Therefore, we can insert valid neighbors in $\Gamma_\pi(u)$ only at an angular distance of more than $\theta/2$ to an existing neighbor. Then the maximum node degree of v is $\lfloor 4\pi/\theta \rfloor$. This is the limit for the number of violating edges and this limit can be reached in the worst case: The example in the

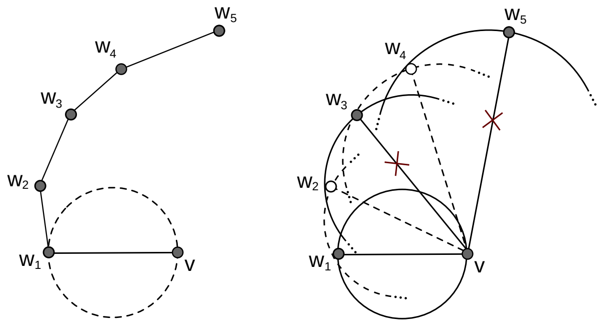


Figure 5: *Gabriel graph (left) and distance-ordered neighborhood (right) with hidden nodes (white) and violating edges (marked with x)*

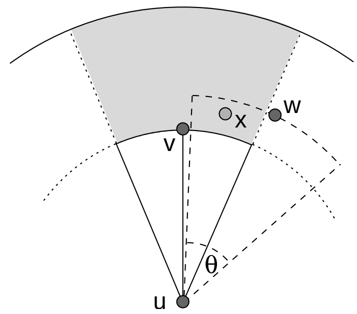


Figure 6: *A proximity region containing a sector bounds the number of violating edges (Theorem 3)*

figure shows that for a pair of nodes with overlapping proximity regions there can always be a hidden node x, with higher rank than v and $v \in N(u,x)$ and $x \in N(u,w)$, that renders (u,w) a violating edge. ■

This theorem shows that we can limit the number of violating edges by choosing an appropriate proximity region. The relative neighborhood graph fulfills this criterion.

Theorem 4 *A distance-ordered relative neighborhood contains at most 4 violating edges.*

Proof: The RNG lune contains a circular sector of $\theta < 120°$. From this fact and Theorem 3 follows the result. ■

Corollary 2 *The beaconless RNG construction with a distance-based delay function requires a constant number of protests in the worst case.*

However, the proximity region of the RNG is quite large, such that more edges are forbidden than in the Gabriel graph. The RNG has (length/power) stretch factor $\Theta(n)$, the Gabriel Graph only $\Theta(\sqrt{n})$ (both are not hop-spanners) [BDEK06].

4.5 Relevance of Protest Messages

We have seen that in the presence of hidden nodes edges can be created that violate the proximity condition. Therefore it is necessary to allow hidden nodes to protest against the selection of a neighbor. One might ask if there is any delay function or any practical subgraph construction that favors only the valid neighbors. Unfortunately this is not the case.

Theorem 5 *No undirected, planar and connected proximity graph can be constructed without protests.*

Proof: Consider the scenario in Figure 7 as a counterexample: Node w is located in the suppression region of v, v is suppressed by u, but w is not suppressed by u. When considering the suppression region for arbitrary proximity graphs (that are undirected, planar and connected), the region is at least the suppression region of the Gabriel graph and at most the suppression region of the RNG. This follows from Lemmata 1 and 2. Therefore, region A is part of the suppression region of v and region B is not a suppression region of u for all considered proximity graphs. Now we build the ordered-neighborhood of x for all permutations of u, v, w.

insertion order π hidden nodes in ()			neighborhood $\Gamma_\pi(x)$	immediate protest	protest of hidden nodes
u	(v)	w	{u,w}		v
u	w	(v)	{u,w}		v
v	u	(w)	{u}	u	
v	(w)	u	{u}	u	
w	u	(v)	{u,w}		v
w	v	u	{u}	v, u	

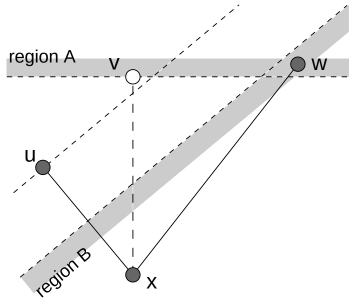


Figure 7: *Hidden node scenario for Theorem 5*

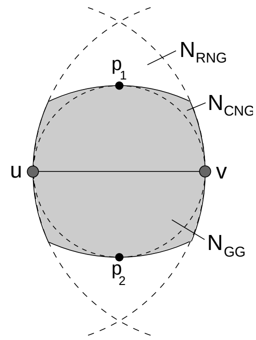


Figure 8: *The circlunar neighborhood with RNG lune and Gabriel circle*

We can see from the table, that regardless of the insertion order, there is always a protest, either because the inserted node immediately knows that it violates the proximity graph condition, or because of a hidden node that protests later. ■

5 The Circlunar Neighborhood Graph

For the beaconless subgraph construction we want to preserve as much edges as possible, bound the number of protests and obtain a planar graph. The planarity can be achieved by including the Gabriel circle in the proximity region. Protests can be bounded by including a circular sector. The larger the angle of the sector, the smaller the maximum node degree, but this also cancels more edges. Therefore, we propose the Circlunar Neighborhood Graph (CNG) as an alternative to Gabriel graph and RNG. It is a planar graph with constant degree; it's proximity region is only a small enhancement of the Gabriel circle and the proximity condition can be tested with 1-hop-knowledge and simple arithmetics.

Definition 6 *The* circlunar neighborhood $N_{CNG}(u,v)$ *of two nodes u and v is given by the intersection of four disks of radius* $|uv|$ *centered at the corners of a square of which* (u,v) *is the diagonal (cf. Fig. 8).*

The cirlunar neighborhood graph contains an edge (u,v) if and only if $N_{CNG}(u,v)$ is empty:

Definition 7 *The* circlunar neighborhood graph *of a node set V contains an edge* (u,v) *iff* $\forall w \in V, w \neq u, v$: $|uv| < \max\{|uw|, |vw|, |p_1, w|, |p_2, w|\}$.

5.1 Properties of the Circlunar Neighborhood Graph

The CNG has a strong relation to Gabriel graph and relative neighborhood graph and inherits planarity and connectivity.

Theorem 6 *The circlunar neighborhood graph of a node set V is planar and connected, if the unit disk graph of V is connected.*

Proof: Follows from the shape of the proximity region and Lemmata 1 and 2. ■

The CNG inherits also a disadvantage from the RNG, namely the unbounded spanning ratio of $\Theta(n)$ (maximum ratio of shortest path in CNG over shortest path in the original graph). One can construct the same lower bound example ("RNG tower" [BDEK06]) for the CNG. In other words, when using the CNG planarization, the maximum detour is unbounded in the worst-case.

In the average case, the CNG has an expected node degree of 3.6 and is thus sparser then the Gabriel graph and denser than the RNG, as the following theorem shows. Table 2 summarizes these results (cf. [Dev88]).

Graph	Exp. degree [Dev88]	Max. degree	Spanning ratio [BDEK06]
RNG	2.558	5	$\Theta(n)$
CNG	3.598	14	$\Theta(n)$
GG	4.000	$n-1$	$\Theta(\sqrt{n})$

Table 2: *Properties of RNG, CNG and GG*

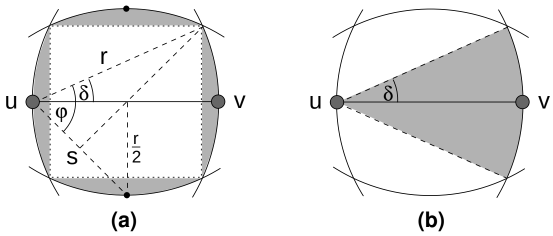


Figure 9: *Illustrations for Theorem 7*

Theorem 7 *The circlunar neighborhood graph has an expected node degree of approx. 3.6.*

Proof: Following the considerations in [Dev88], we can derive the expected degree of the CNG from the ratio of the circle $C(u,|uv|)$ and the area A of the proximity region $N_{CNG}(u,v)$. A can be composed of 4 segments (shaded area in Fig. 9a) and the remaining square. The area of one segment is $A_{seg} = \frac{1}{2}r^2(\theta - \sin\theta)$. The angle $\theta := 2\delta$ is calculated as follows: $s = \sqrt{2(r/2)^2}$. $\cos\varphi = \frac{s/2}{r} = \frac{1}{4}\sqrt{2}$. $\delta = \varphi - 45° \approx 24.3$. The area of the square is $A_{\square} = (2\,r\sin\delta)^2$. Plugging in $\theta = 2(\arccos(\frac{1}{4}\sqrt{2}) - \frac{\pi}{4})$ and adding the area of the square and four segments gives the total area: $A = A_{\square} + 4\,A_{seg} \approx 0.873\,r^2$. This gives an expected degree of $E[d] = C(u,|uv|)/A \approx 3.598$. ■

Besides the theoretical considerations, we performed simulations on 200 random unit disk graphs with 100 nodes for network densities (avg. number of neighbors) between 4 and 12. Measurements of the spanning ratio show that the CNG is closer to the Gabriel graph than to the RNG: The hop spanning ratio of the CNG is only 5%-7% larger than in the Gabriel graph, while the RNG's spanning ratio is 36%-61% larger (see Fig. 10).

5.2 Beaconless construction

The CNG enables a beaconless planar subgraph construction with a constant number of protests as the following theorem shows.

Corollary 3 *A distance-ordered neighborhood in the CNG has at most 13 violating edges.*

Proof: This follows from Theorem 3. One can show that the circlunar neighborhood contains a circular sector of $\approx 48,6°$. Plugging this into Theorem 3 gives the result. ■

5.3 Face Routing on the Circlunar Neighborhood Graph

The circlunar neighborhood graph has the structural graph properties that are necessary to guarantee recovery. The following graph property holds for the Gabriel graph (Lemma 1 in [FS06]) and can be shown analogously for the CNG.

Lemma 3 *For any edge (u,v) crossing the s-t-line connecting source s and destination t in the circlunar neighborhood graph, at least one of the end points u or v is closer to the target than s.*

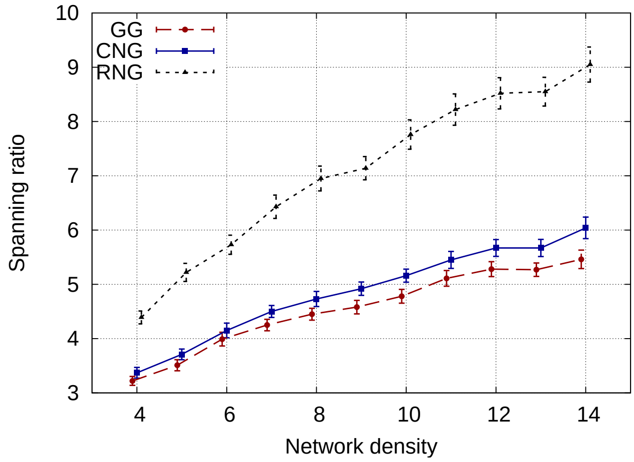


Figure 10: *Spanning ratio of RNG, CNG and GG (average with 95% confidence error for 200 random graphs, 100 nodes)*

Proof: As the circlunar neighborhood contains the Gabriel circle, the Gabriel circle over (u,v) neither contains s nor t. It follows that $\angle usv$ and $\angle utv$ are less than $\pi/2$. Since the sum of the angles of the quadrangle $usvt$ is 2π, at least one of the angles $\angle sut$ or $\angle svt$ is greater than $\pi/2$. This implies that at least one of the nodes u or v is closer to t than s. ■

For guaranteed delivery, face routing on the planar subgraph has to provide progress towards the destination. This is shown by the following theorem (cf. Corollary 2 of [FS06]).

Theorem 8 *Let s and t be nodes in a circlunar neighborhood graph. When starting at s, face routing will always find a node v that satisfies $|vt| < |st|$.*

Proof: The CNG is planar and from Lemma 5 in [FS06] follows that face routing will always find an edge intersecting the s-t-line. With Lemma 3 we can conclude, that one of the edge's end points satisfies $|vt| < |st|$. ■

6 Angular Relaying

Angular Relaying is a beaconless face routing strategy, which can be used as a method for recovery from local minima. While BFP works independent of the routing protocol, Angular Relaying needs the information of the previous hop and the recovery direction (right-hand or left-hand). It is based on an angle-based delay function to determine a candidate for the next hop which is used in combination with the select-and-protest method for avoiding crossing edges. Here, we use the Gabriel graph condition as planarization criterion.

By using an angle-based delay function the first neighbor in counter-clockwise order is selected. Other approaches, such as NB-FACE, the clockwise relaying approach in an earlier version of BLR [HB03], or the Bypass method of PSGR are also based on an angle-based function, but they either cannot guarantee delivery or the complete neighborhood is involved in the message exchange. An simple angle-based delay function has the following form:

$$t(\theta) = \frac{\theta}{360^\circ} \cdot t_{\max} \tag{2}$$

The angle θ can be considered in clockwise or counter-clockwise order, depending on the traversal direction (left-hand or right-hand). Selecting a candidate by this function is not sufficient to guarantee delivery,

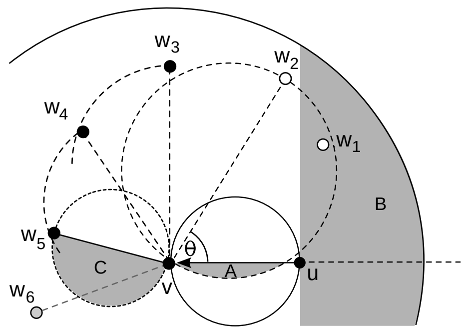


Figure 11: *Angular Relaying: w_1 and w_2 are invalid, w_3 is selected, w_4 and w_5 protest. Finally, w_5 is the next hop.*

because it is not necessarily a neighbor of the forwarder in the Gabriel subgraph. Therefore, we use protest messages to prevent crossing links. This is similar to the protest phase used in the BFP algorithm.
The Angular Relaying algorithm consists of two phases:

1. Selection phase After receiving a packet from the previous hop u, the forwarder v sends an RTS (including previous hop u and own position) and sets its timer to $t_{\max}$. Every candidate w sets its timer $t(\theta)$ using the angular distance $\theta = \angle uvw$ to the previous hop. Candidates answer with a CTS in counter-clockwise order, according to the delay function. We allow candidates to respond, if they have the previous hop in the Gabriel circle (i.e. nodes in region B in Fig. 11). These nodes answer with an "invalid CTS", because they violate the Gabriel graph condition, but other nodes should be aware of their existence. Otherwise they would be hidden and need a chance to protest later. After the first candidate w answers with a valid CTS, the forwarder immediately sends a SELECT message announcing that w is the first selected node. All candidates with pending CTS answers cancel their timers.

2. Protest phase After the selection of the first candidate, the protest phase begins. The forwarder starts its protest timer that covers only the time when protests can occur, which is $t_{pr} = t(\frac{\pi}{2}) = \frac{1}{4}t_{\max}$ for the Gabriel graph. Now, no further CTS answers are allowed. Instead, each candidate x sets a new timer $t(\theta)$ that determines the order of protests ($\theta = \angle uvx - \angle uvw$). First, only nodes in $N_{GG}(v,w)$ are allowed to protest. If a node x protests, then it automatically becomes the next hop. After that, only nodes in $N_{GG}(v,x)$ are allowed to protest. Finally, if the forwarder's timer expires (i.e. there are no more protests), the data packet is sent to the currently selected (first valid or last protesting) candidate.

Angular Relaying using a simple angle-based delay function (equation 2) is similar to NB-FACE. In NB-FACE the forwarder waits for a time span τ after the first candidate responded, in order to leave room for protests ('NAck'). After that, it sends a message ('Fin') to stop the contention period and select the final candidate. If τ is a constant angle, then the case of cascading protests is not covered; otherwise, if τ spans the whole rotation, then all neighbors respond, even if they are not protesting, and the advantage over the Request-Response approach vanishes. Also the details about how nodes are treated that have the previous hop in their Gabriel circle (region B in Fig. 11) are left open.

6.1 Angular Relaying using a Sweep Curve

The contention process using the angular delay function can be regarded as a rotating sweep line, i.e. a ray from v through u (Fig. 11) that rotates in counter-clockwise order until it hits the first node w. This node will be the next hop, if it is a valid neighbor so far and there are no protests afterwards. Protests are issued by nodes that lie in the Gabriel circle over (v,w) and beyond the sweep line (region C in Fig. 11). Therefore, it makes no sense to use CNG or RNG with Angular Relaying, because the area of possible protesting nodes would be even larger. We also observe, that the protest area grows with the distance of a candidate to the forwarder.

Task of the forwarder v
(after receiving a message from u)

1. next_hop = u
   send the RTS
2. set timer $T_1 = t_{\max}$
   while T_1 is not expired
   upon reception of a valid CTS from node w:
   send SELECT(w)
   next_hop = w
   cancel timer: $T_1 = 0$; proceed with step 4
3. on expiration of T_1:
   forward data packet to next_hop
4. set timer $T_2 = t_{\max}/4$ (protest timer)
   while T_2 is not expired
   upon reception of a Protest from node x:
   next_hop = x
   reset timer $T_2 = t_{\max}/4$
5. on expiration of T_2:
   forward data packet to next_hop

Task of a candidate w
(after receiving an RTS from forwarder v, prev. hop: u)

1. valid = $u \notin N(v,w)$
2. set timer $T_1 = t(|vw|, \angle uvw)$
   while T_1 is not expired
   upon reception of a CTS from node x:
   if $x \in N(v,w)$ or $w \in N(v,x)$ then
   valid = false
   upon reception of SELECT(x) from the forwarder:
   next_hop = x, proceed with step 4
3. on expiration of T_1:
   if (valid=true) then send valid CTS
   else send invalid CTS
4. if $w \in N(v$,next_hop$)$ then
   set timer $T_2 = \min\{t_{\max}/4, T_1\}$
   while T_2 is not expired
   upon reception of a protest from another node y:
   next_hop = y, resume step 4
   else halt
5. on expiration of T_2:
   send a Protest (incl. the own position)

Figure 12: *Angular Relaying*

This leads to the question whether another shape of the sweep line could be applied such that closer nodes may respond earlier and the area of possible protesting nodes is reduced. To ensure that the most suitable nodes respond first, the sweep curve must have the following property:

Sweep curve property: For a node w on the sweep curve must hold: If there is another node x ahead of the sweep curve in counter-clockwise order, then either $\angle uvw < \angle uvx$ or x is not a Gabriel neighbor of v.

In other words: If w responds first, then there are no other Gabriel neighbors with smaller angular distance to the previous hop (θ), that could respond later and contradict w being the first neighbor in counterclockwise order (cf. Fig. 15). In order to determine a valid sweep curve (for the Gabriel graph construction), we consider the suppression region for a node w and calculate the positions for which all nodes with smaller θ are suppressed (see Fig. 13). Fulfilling the sweep curve property requires that $x + z \leq z$. For the height of a rectangular triangle holds $y^2 = xz$ and it follows that $x + \frac{y^2}{x} \leq r$, i.e. all nodes on the sweep curve should lie between the straight line and the semi-circle in Fig. 13.

6.2 Correctness of the Angular Relaying Algorithm

Theorem 9 *The Angular Relaying algorithm selects the first edge of the Gabriel subgraph in counter-clockwise order.*

Proof: Let $N^+_{GG}(v,w)$ be the left part of the Gabriel circle of (v,w), which is ahead of the sweep line/curve (region C in Fig. 11). Analogously, let $N^-_{GG}(v,w)$ be the remaining part of the Gabriel circle. For the first selected candidate w holds, that $N^-_{GG}(v,w)$ is empty, because of the following reasons: First, all nodes are allowed to respond, also the invalid ones. That ensures that there are no hidden nodes invalidating w. Thus, w has the smallest angle $\angle uvw$ among the Gabriel neighbors (otherwise another valid neighbor would have responded before). There is only one region that we did not consider yet, namely the part of $N^-_{GG}(v,w)$ beyond $\overline{uv}$ (region A in Fig. 11). But this region is empty, because it is always covered by

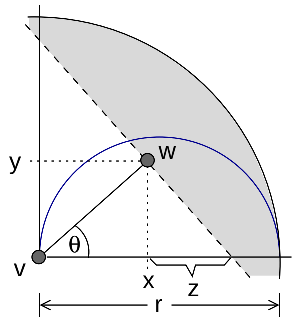


Figure 13: *Suppression region and sweep curve property*

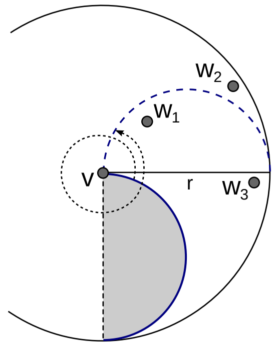


Figure 14: *The general sweep curve needs a $5\pi/2$ turn to cover all nodes.*

$N_{GG}(u,v)$; otherwise, (u,v) would not be an edge of the Gabriel graph. For similar reasons, $N^{-}_{GG}(v,w)$ is empty for nodes that protest in the second phase. Protesting nodes are automatically selected as tentative next hop. For the currently selected node w holds (invariant of the algorithm): If $N^{+}_{GG}(v,w)$ is empty, then w is the Gabriel neighbor of v with the smallest angle $\angle uvw$. This follows from the sweep curve property (no node with smaller angle responds later) and the considerations above. The algorithm terminates if the forwarder's timer expires. Then $N^{+}_{GG}(v,w)$ is empty, because there are no further protests. If a part of the Gabriel circle intersects with the radial line from v through u, then this part lies within the Gabriel circle over (u,v) and therefore, this region is also empty. ■

6.3 Sweep Curve Functions

In general, a sweep curve function f, which describes a point on the sweep curve by angle θ and distance $f(\theta)$, has to be monotonic and fulfill the following conditions:

(1) $f(0) = 1$

(2) $f(\pi/2) = 0$

(3) $0 \le f(\theta) \le \cos(\theta)$ for $\theta \in [0..\pi/2]$

The delay function is derived from the inverse of f and has the following general form:

$$t(d,\theta) = \frac{\theta - f^{-1}(d/r) + \pi/2}{5\pi/2} \cdot t_{\max} \tag{3}$$

In order to calculate the expected protest area, we consider a fixed angle θ and calculate the area enclosed by the sweep curve and the Gabriel semi-circle (shaded lune in Fig. 15).

$$A_P(\theta) = \frac{1}{2}\pi\left(\frac{f(\theta)}{2}\right)^2 - \int_{\theta}^{\frac{\pi}{2}}\int_0^{f(\theta)} r\,dr\,d\varphi \tag{4}$$

$$= \frac{\pi}{8}f(\theta)^2 - \frac{1}{2}\int_{\theta}^{\frac{\pi}{2}} f(\varphi)^2 d\varphi \tag{5}$$

The probability of a node in distance r is $\frac{2\pi r}{\pi} = 2r$ in the unit circle. Thus, the expected protest area is given by

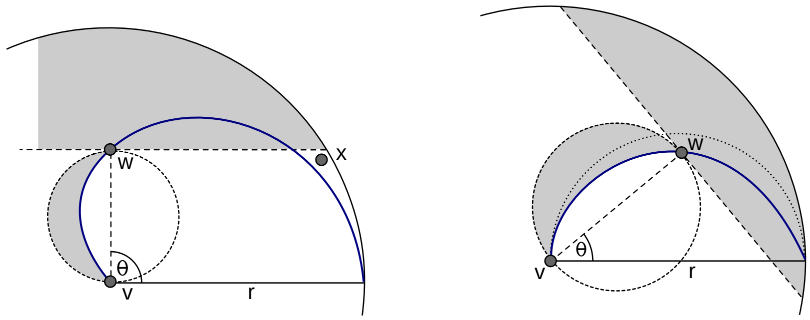
$$\mathrm{Exp}[A_P] = \int_0^1 A_P(f^{-1}(r)) \cdot 2r\,dr \tag{6}$$

The optimal curve that minimizes the protest area is a logarithmic spiral, which has the form $a \cdot e^{b\theta}$. However, these spirals are not valid sweep curves, because $e^{\theta} > 0$ violates condition 2 (cf. Fig. 15a). Valid

Sweep curve function	$f(\theta)$	$f^{-1}(r)$	Expected protest area
sweep line	—	$\pi/2$	0.1963
semi-circle	$\cos(\theta)$	$\arccos(r)$	0.0982
Archimedean spiral, $c = 1.259$	$1-\left(\frac{2}{\pi}\theta\right)^c$	$\frac{\pi}{2}(1-d)^{1/c}$	0.0897
logarithmic spiral*	$e^{-\frac{2}{\pi}\theta}$	$\frac{\pi}{2}\ln(d)$	(0.0531)

*) not a valid sweep curve

Table 3: *Sweep curve variants*



(a) When using an arbitrary spiral, there can be a valid Gabriel neighbor x beyond the sweep curve with smaller angle θ and larger delay

(b) The optimum Archimedean spiral that fulfills the sweep curve property. The dotted line indicates the semi-circle.

Figure 15: *Angular Relaying with different sweep curves*

sweep curves are the semi-circle ($f(\theta) = \cos(\theta)$) or some Archimedean spirals. An Archimedean spiral has the general form $a + b\theta^c$. We use the form $f(\theta) = 1 - \left(\frac{2}{\pi}\theta\right)^c$, which fulfills conditions 1 and 2. The exponent c determines the shape of the curve and whether condition 3 can be fulfilled. Condition 3 requires $1 - \left(\frac{2}{\pi}\theta\right)^c \leq \cos(\theta)$ follows that $c \leq \frac{\log(1-\cos(t))}{\log(\frac{2}{\pi}\theta)}$, i.e. $c \leq 1.56$ for $\theta \in [0..\pi/2]$. Using equation 6, the expected protest area is minimized, if $c \approx 1.259$ (numeric evaluation). With this function, we can reduce the expected protest area by more than a factor of 2 compared to the sweep line. Table 3 summarizes the results for different functions.

7 Simulations

We performed simulations of BFP and Angular Relaying on 500 random graphs with 100 nodes for network densities (i.e. avg. number of neighbors) ranging from 4 to 12. Messages are sent from the leftmost to the rightmost node using GFG routing. The greedy part is performed by a beaconless greedy scheme using RTS/CTS, the face routing part is performed by BFP on different subgraphs or by Angular Relaying using sweep line and sweep curve. We use a simplified MAC layer model assuming uniform transmission radii and no collisions. We measure the number of messages used for each route. In order to obtain a fair and consistent measure for different routing paths and subgraphs, the values are normalized, i.e. divided by the length of the shortest path (number of hops) in the original unit disk graph.

Beaconless Forwarder Planarization

The results for the number of protests (Fig. 17) and for the overall message complexity (Fig. 16) show a gap between Gabriel graph and RNG. The inferior performance of RNG is due to the long detours caused by this planarization method. The CNG reaches the good performance of the Gabriel graph, while

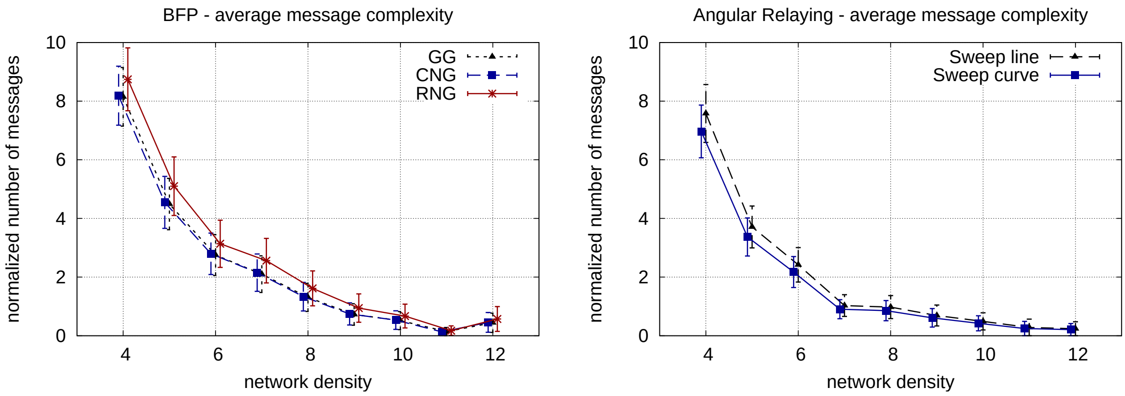


Figure 16: *Message complexity of BFP and Angular Relaying (average with 95% confidence error for 500 random graphs, 100 nodes)*

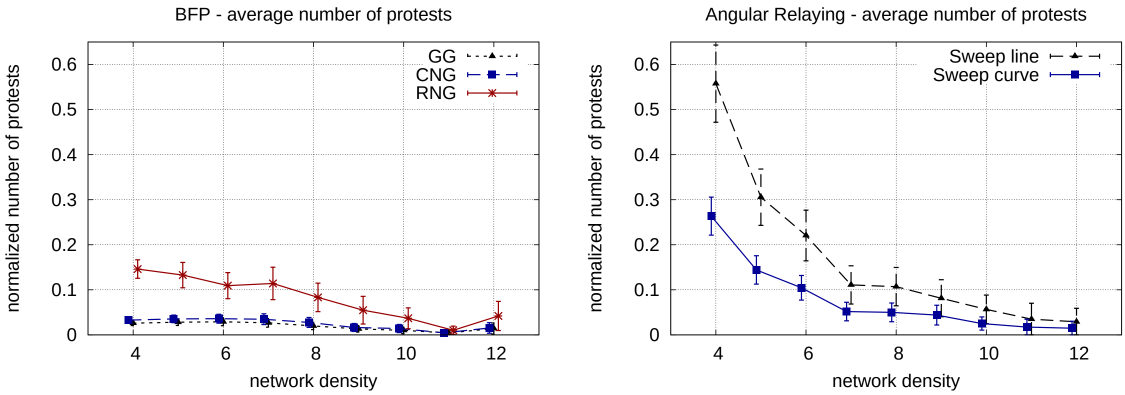


Figure 17: *Protests of BFP and Angular Relaying (average with 95% confidence error for 500 random graphs, 100 nodes)*

guaranteeing a worst-case bound for the number of protests, which is not possible when using the Gabriel graph planarization.

Angular Relaying

When using sweep curve instead of sweep line, we observe a reduction of the number of protests by more than a factor of 2 on average. This corresponds to our theoretical results showing a reduction of the expected protest area by the same factor. The reduction of protests leads to an overall message reduction of 11% on average. Note, that greedy routing is used for large parts of the routing path.

We can also observe, that BFP uses less protest messages than Angular Relaying. But that does not imply that BFP is more efficient, because some nodes that send a protest in Angular Relaying would send a CTS when using BFP. This is reflected in the overall message complexity, where Angular Relaying uses less messages. However, BFP constructs a complete local subgraph, Angular Relaying determines only the next hop.

8 Conclusion

We have presented two solutions for the beaconless recovery problem and introduced a theoretical framework to analyze the message complexity of beaconless face routing algorithms. Both solutios follow the Select-and-Protest principle, which is a message-efficient approach for beaconless face routing and enables a completely reactive georouting with guaranteed delivery. We could improve the message complexity by introducing a new planar subgraph construction and new delay functions. Further improvements could be achieved by storing and using overheard transmissions in an RTS cache. Also, BFP as proposed works independent of the routing algorithm and can be improved by a closer interaction, where routing decisions are made before the protest phase. Future research includes extensions to handle discrete timeouts and collisions when using a realistic MAC layer.

References

[BDEK06] Prosenjit Bose, Luc Devroye, William Evans, and David Kirkpatrick. On the spanning ratio of gabriel graphs and beta-skeletons. *SIAM Journal on Discrete Mathematics*, 20(2):412–427, 2006.

[BGM04] Prosenjit Bose, Joachim Gudmundsson, and Pat Morin. Ordered theta graphs. *Computational Geometry*, 28(1):11–18, May 2004.

[BHSS03] Brian Blum, Tian He, Sang Son, and John Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. Technical Report CS-2003-11, University of Virginia, USA, April 2003.

[BMSU99] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *3rd int. workshop on Discrete algorithms and methods for mobile computing and communications (DIALM '99)*, pages 48–55. ACM Press, 1999.

[CGK+06a] Mohit Chawla, Nishith Goel, Kalai Kalaichelvan, Amiya Nayak, and Ivan Stojmenovic. Beaconless position based routing with guaranteed delivery for wireless ad-hoc and sensor networks. In *1st IFIP Int. Conference on Ad-Hoc Networking*, pages 61–70. Springer, August 2006.

[CGK+06b] Mohit Chawla, Nishith Goel, Kalai Kalaichelvan, Amiya Nayak, and Ivan Stojmenovic. Beaconless position-based routing with guaranteed delivery for wireless ad hoc and sensor networks. *ACTA AUTOMATICA SINICA*, 32(6):846–855, November 2006.

[Cim92] Robert J. Cimikowski. Properties of some euclidean proximity graphs. *Pattern Recognition Letters*, 13(6):417–423, 1992.

[CV07] D. Chen and P. K. Varshney. A survey of void handling techniques for geographic routing in wireless networks. *IEEE Communications Surveys and Tutorials*, 2007.

[Dev88] L. Devroye. The expected size of some graphs in computational geometry. *Computers & Mathematics with Applications*, 15(1):53–64, 1988.

[Fin87] Gregory G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, University of Southern California, March 1987.

[FS06] Hannes Frey and Ivan Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *12th Annual Int. Conference on Mobile Computing and Networking (MobiCom'06)*, 2006.

[FWMH03] Holger Füßler, Jörg Widmer, Martin Mauve, and Hannes Hartenstein. A novel forwarding paradigm for position-based routing (with implicit addressing). In *IEEE 18th Annual Workshop on Computer Communications (CCW 2003)*, pages 194–200, October 2003.

[GGH+01] Jie Gao, Leonidas J. Guibas, John E. Hershberger, Li Zhang, and An Zhu. Geometric spanner for routing in mobile networks. In *2nd Symposium on Mobile Ad Hoc Networking and Computing*, pages 45–55, October 2001.

[GS69] K. Ruben Gabriel and Robert R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18 (3):259–278, 1969.

[HB03] Marc Heissenbüttel and Torsten Braun. A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks. In *3rd IEEE Workshop on Applications and Services in Wireless Networks*, pages 197–209, 2003.

[HBBW04] Marc Heissenbüttel, Torsten Braun, Thomas Bernoulli, and Markus Wälchli. BLR: Beacon-less routing algorithm for mobile ad-hoc networks. *Computer Communications*, 27(11):1076–1086, July 2004.

[JT92] Jerzy W. Jaromczyk and Godfried T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. of the IEEE*, 80:1502–1517, 1992.

[LCW02] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *21st Annual IEEE Conference on Computer Communications (INFOCOM)*, 2002.

[Li03] Xiang-Yang Li. Approximate MST for UDG locally. In *9th Annual Int. Conference on Computing and Combinatorics*, pages 364–373, 2003.

[LSW04] 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.

[NOH06] Minami Narasawa, Masakazu Ono, and Hiroaki Higaki. NB-FACE: No-beacon face ad-hoc routing protocol for reduction of location acquisition overhead. In *7th Int. Conf. on Mobile Data Management (MDM'06)*, page 102, 2006.

[TK84] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, March 1984.

[WT05] Matthias Witt and Volker Turau. BGR: Blind geographic routing for sensor networks. In *3rd Workshop on Intelligent Solutions in Embedded Systems (WISES'05)*, pages 51–61, 2005.

[XLXM05] Yingqi Xu, Wang-Chien Lee, Jianliang Xu, and Gail Mitchell. PSGR: Priority-based stateless geo-routing in wireless sensor networks. In *2nd IEEE Int. Conference on Mobile Ad Hoc and Snsor Systems (MASS'05)*, 2005.

[Zor04] Michele Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *IEEE Int. Conference on Communications (ICC 2004)*, pages 3481–3485, 2004.