A New Traversal Scheme for Georouting and Boundary Detection in WSNs

Stefan Rühruр [(1)] Ivan Stojmenović [(2)]

Technical Report
University of Ottawa
August 2010

Abstract Knowing or determining the boundaries in wireless sensor networks is most important for geographic routing. Geographic greedy strategies, which try to bring a message closer to the destination in each step, may fail at so-called local minima. Local minima exist at the outer network boundaries or at void regions inside the network. The detection and traversal of these boundaries is a key to overcome greedy routing failures, but also to detect and avoid regions of network congestion, node failures, or power depletion.

Existing localized georouting algorithms follow boundaries of void regions by right-hand traversals and require a planar subgraph construction, which removes edges that could lead to routing loops. In this paper we show that this planarization is not required, i.e. that network boundaries can be found without explicit removal of intersections. Our protocol follows extends the contention-based routing algorithm in [24]. It works fully reactive and does not require other nodes than the boundary nodes to actively participate in the message exchange. Simulations show that neighbor selection scheme provides shorter boundary traversal paths than existing approaches. It also improves known boundary detection techniques, which still rely on a priori neighborhood information.

(1) OFFIS – Institute for Information Technology, Oldenburg, Germany, stefan.ruehrup@offis.de
(2) School of Information Technology and Engineering, University of Ottawa, Canada, ivan@site.uottawa.ca

1 Introduction

A wireless sensor network is a set of small devices, called sensor nodes, which are able to sense, process, and communicate data. Potential application areas (e.g. environmental monitoring, precision agriculture, home automation) are manyfold and real sensor networks have been deployed in some of these areas in the recent years. One important issue among the various design challenges in developing sensor nodes is to ensure network-wide wireless communication among a multitude of nodes, each with limited transmission range and limited processing capabilities while using a shared a communication medium.

In this paper we focus on contention-based (also called beaconless) geographic routing, a fully reactive multi-hop routing scheme based on position information of the nodes. The main advantage of such schemes is that routing decisions require only local information. Thus the overhead of exchanging topology or routing information can be minimized, which enables scalable solutions for large-scale sensor networks. The first localized geographic routing algorithms were greedy strategies, which let each node forward a packet to one of its direct neighbors that is closer to the (previously known) destination position than itself. This greedy forwarding scheme works well in dense networks, but experiences routing failures in more realistic, non-uniform node placements, where it suffers from so-called local minima, i.e. nodes with no direct neighbor closer to the destination than themselves.

For such cases, face routing [3] became an established technique to route packets using geographic information: Face routing guides packets along faces of the network communication graph and guarantees delivery when applied on a planar subgraph. These faces enclose void regions in the network on whose border one can find local minima. Therefore it was observed by Fang et a. [7] that it is essential to determine the boundary of a void region rather than to construct a planar subgraph.

Problem formulation: The problem is to identify a boundary of a node set, that encloses all nodes of a connected component and contains all local minimum nodes on its border. This has to be achieved by a fully reactive and localized algorithm, which cannot rely on knowledge of the 1-hop neighborhood in advance. Positions of 1-hop neighbors have to be requested explicitly by message exchange.

A recent contention-based routing strategy [24] shows that a planar boundary can be constructed in a fully reactive way by excluding nodes incident to possibly intersecting edges. In this paper we show that a boundary does not need to be free of intersections and define a traversal strategy that does not explicitly forbid edge intersections, but avoids those that potentially lead to routing loops. This preserves more edges and leads to fewer hops on average in simulations. The strategy is easy to implement, but it is non-trivial to prove that it indeed produces valid boundaries. We present such proof and show by simulations that boundaries are shorter than the best known reactive approaches.

The paper is organized as follows: In Section 2 we review related work. Section 3 describes the new Twisting Triangle scheme and its integration in the RS algorithm. Its correctness proof follows in Section 4. In Section 5 we show theoretical properties. Simulation results are presented in Section 6. A summary in Section 7 concludes the paper.

2 Related Work

The algorithm presented here provides a method to identify outer network boundaries or boundaries of void regions inside a wireless sensor network by a reactive contention-based protocol, which is related to and relevant for geographic routing and boundary detection. For the two purposes similar algorithms have been developed, all with the same goal to find or traverse the boundary using the smallest number of nodes.

2.1 Geographic routing

Important achievements in the evolution of geographic routing are the first greedy algorithms in the 1980s such as the *greedy method* [10], and face routing in the late 1990s [3], which are still considered the basic building blocks of many georouting algorithms. The greedy method, also called greedy forwarding, tries to minimize the distance to the destination in each step, and thus makes locally optimal decisions. The main limitation of greedy forwarding is that it fails in local minima, i.e. when reaching a node that has no neighbor closer to the destination than itself. Those nodes exist at the network boundaries such as at the border of void regions. Face routing [3] is able to recover from such situations: it performs traversals of faces of the network communication graph and guarantees delivery when applied on its own or in combination with greedy forwarding. Face routing requires a planar subgraph of the communication graph, which can be constructed locally by using the Gabriel graph (GG) [15]. A planar graph is a sufficient condition to guaranteed delivery of face routing [13]. However, local planarization strategies forbid edges that could possibly lead to routing loops, but they might also forbid useful edges and thus increase the traversal length. This problem of long traversal paths has been approached by strategies on a microscopic scale and on a macroscopic scale: Shortcutting techniques shorten paths by including 2-hop-neighborhood information [5] in order to select better next hop neighbors. Approaches to optimize the asymptotic behaviour of geographic such as [21] introduce additional techniques on a macroscopic scale, but still rely on face routing or similar methods of boundary traversals. A comprehensive overview of planar graph routing is given in [12, 4].

2.2 Contention-based georouting

The first fully reactive geographic routing schemes were developed independently by several researchers and appeared 2003 under the names beacon-less routing (BLR) [17], contention-based forwarding (CBF) [14], implicit geographic forwarding (IGF) [1]. They belong to greedy forwarding algorithms and have in common that the next hop is selected in a timer-based contention mechanism, where the most suitable candidates, which maximizes the advance towards the target, is favored. When integrated into a RTS-CTS scheme, the contention process works as follows:

After broadcasting a request-to-send (RTS) by the forwarder, the candidates start a timer. The timeout interval or *delay* of a candidate w depends directly on its advance a, which is $a := |vT| - |wT|$, where v is the forwarder position and T the target. The larger the advance of a candidate, the smaller is its delay. Upon timer expiry, a candidate answers with a CTS and immediately obtains the DATA message from the forwarder. These messages are overhead by other candidates, which cancel their scheduled responses. By this scheme the forwarder can directly identify the next hop without knowing other neighbors.

The delay function for a contention-based greedy algorithm is a function of the advance a and returns the length of the timeout interval. It is defined as

$$t_{\text{greedy}}(a) = \frac{a}{r} \cdot t_{\max}$$

where r is the transmission range, and $t_{\max}$ the maximum timeout, which defines the length of the contention period.

This scheme was extended in previous works to allow a contention-based recovery. The Angular Relaying algorithm [20] uses a delay function that depends on the angle between previous hop, forwarder and next hop, such that the first hop in counter-clockwise order responds first. Further protest messages by other nodes ensure that edges of the Gabriel graph remain. The continuation of this idea led to the Rotational Sweep algorithm [24] that also uses a delay based on an angular ordering. This algorithm directly selects neighbors of the so-called α-shape of the network (with $\overline{\alpha} = r/2$, see Fig. 1) and does not need protest messages. α-shape traversals are subsequences of Gabriel subgraph traversals and thus provide shorter recovery paths.

Table 1 gives an overview of different subgraph constructions.

2.3 Boundary detection

The detection of a boundary and its traversal is essential to geographic routing, especially when recovering from a greedy forwarding failure. It is also useful to avoid regions of network congestion, node failures, or power depletion. Further applications are described in [8].

The boundary of a network consists of its outer boundary and boundaries around void regions in the network (cf. Fig.23). The problem is to determine a closed curve around connected component that separates its contained nodes from voids or unreachable regions. The main problem here is to define criteria for nodes on the boundary and how they are connected, such that these criteria lead to an intersection-free path in the unit disk graph that represents the aforementioned closed curve.

In a seminal work on boundary detection for sensor networks Fang et al. defined such algorithm, which they called BoundHole [7]. They specified a criterion for nodes on the boundary, the so-called stuck nodes: If for a node u there is a location outside its transmission radius where u is closer to than any of its 1-hop neighbors, then u is called a (strongly) stuck node. These nodes can identify themselves by applying the aforementioned rule and with knowledge of their 1-hop neighborhood. Stuck nodes start the BoundHole algorithm, that lets each node choose the next edge in counterclockwise order after the previous edge, while leaving out previously visited nodes. This leads to a traversal around a void region. BoundHole does not forbid edge intersections on the traversal. These intersections can only be removed if a path intersects itself, or if other stuck nodes on the same boundary also start the BoundHole algorithm. BoundHole can successfully identify boundaries around void regions. However, as the edge intersection problem cannot be resolved locally, the approach cannot turned into a fully reactive solution.

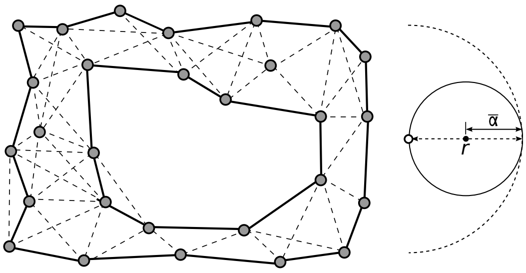


Figure 1: *Unit disk graph with radius r and negative α-shape (solid edges) with $\overline{\alpha} = r/2$ of a point set.*

Important graph-theoretical foundations of boundary detection were laid by Edelsbrunner et al. [6] in the 1980s with the definition of α-shapes. α-shapes connect nodes on a so-called α-hull, a generalization of the convex hull (see Fig. 1). Among this class, negative α-shapes are of particular importance for boundary detection, as they provide a "tighter fit" to the enclosed point set than the convex hull. Informally, a negative α-shape consist of nodes that are touched by a fixed disc of radius $\overline{\alpha}$ that is moved around a point set (cf. Definitions in [6]) and the respective edges.

There are algorithms for sensor networks that yield α-shapes: A localized α-shape construction based on a restricted Delaunay triangulation was shown in [9] for boundary detection. Liu and Feng proposed an algorithm for geographic routing that uses a 'rolling ball' scheme, which actually performs an α-shape traversal, without calculating the Delaunay triangulation [23]. In our previous work we defined a contention-based algorithm that yields α-shapes as well and works without local neighborhood knowledge [24]. All algorithms use a disc radius of $\overline{\alpha} = r/2$, where r is the unit disk graph radius.

In contrast to boundaries found by the BoundHole algorithm, α-shapes are planar boundaries, where paths never intersect themselves and do not need an explicit intersection detection. However, simulations on random graphs in [8] show that BoundHole boundaries are by 14% shorter (in terms of number of nodes) than boundaries of a restricted Delaunay triangulation (where edges cannot exceed the unit disk radius), which is in turn a supergraph of the α-shape. In this paper we close the gap between the two approaches and present a localized algorithm, that provides shorter boundaries than α-shape traversals and directly avoids critical intersections.

3 Rotational Sweeps and Twisting Triangles

3.1 The RS algorithm

The Rotational Sweep (RS) algorithm was presented in [24] and describes a strategy to find the next hop on the network boundary by a contention-mechanism. It is started at a local minimum node and identifies the next hop in counter-clockwise order. The selection of the next hop candidate is controlled by a delay function: After the forwarder (i.e. the node holding

Subgraph construction	Neighborhood-based algorithm	Fully reactive algorithm
Gabriel graph, RNG	Face routing [3]	Angular Relaying [20]
α-shape (for $\overline{\alpha} = r/2$)	Greedy anti-void routing [23], Local α-shape [9]	RS Routing (Sweep circle) [24]
Delaunay Triangulation (localized)	[16], [22]	(unknown)
Intersection removal	Direct Planarization [11]	(unknown)

Table 1: *Overview of localized subgraph constructions for planarization, georouting, and boundary traversal*

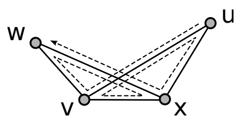


Figure 2: *Detour on crossing links*

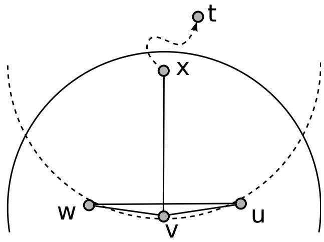
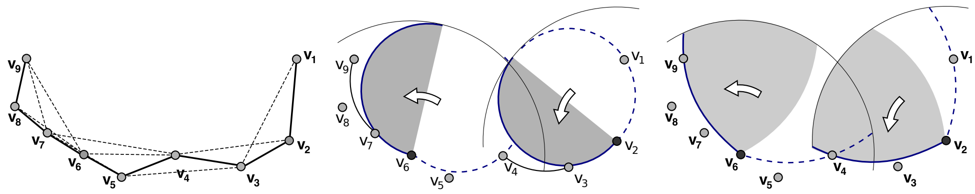


Figure 3: *Routing loop caused by crossing links. Node u cannot determine locally if uw is intersected.*



(a) Unit disk graph of a point set and Gabriel subgraph (solid edges)

(b) Sweep Circle Traversal

(c) Twisting Triangle Traversal

Figure 4: *Unit disk graph and traversals. The boundary of the Gabriel subgraph contains all nodes $v_1, \ldots, v_9$. The Sweep Circle skips node v_8 in this sequence, the Twisting Triangle traversal skips v_3, v_5, v_7 and v_8 and visits only v_1, v_2, v_4, v_6, v_8.*

a message) broadcasts a Request to Send (RTS), all neighboring nodes, the so-called candidates, schedule to respond by a Clear to Send (CTS) after a certain delay. The node with the shortest delay replies first, upon which the forwarder immediately sends the DATA packet. Other candidates with pending replies cancel their scheduled transmissions. What is crucial to the algorithm is the delay function, which depends on the relative location of candidate, forwarder and previous node (whose coordinates have to be included in the RTS and CTS control messages). The delay function of the RS algorithm can be visualized geometrically: It works like a geometric curve which is rotated around the forwarder; once it hits a node, this node becomes the next hop on the boundary. The delay function itself gives the time when the node is hit by the curve. In [24] the delay function represents a rotating semi-circle and was therefore called Sweep Circle; it yields traversals of the α-shape with $\overline{\alpha} = r/2$. For convenience the pseudocode of the RS algorithm is repeated here in Table 2.

The Rotational Sweep Algorithm
Variables: previous hop u, current node v, neighbor w, target T

Action of forwarder v

```
on reception of a traversal message from node u do
  send RTS[u,v] and wait for CTS by neighbor w
    (neighbor w calculates the delay using the RS algorithm)
  forward DATA message to w
od
```

Action of candidate w

```
on reception of RTS[u,v] do
  t = t(|vw|, ∠uvw)
  if (t > 0 and t < ∞)
    wait for t time steps
    if no other candidate has responded before
      send CTS packet to v
  fi
od
```

Table 2: *The Rotational Sweep Algorithm [24] with a minor modification w.r.t. the new delay function.*

3.2 Twisting Triangle and its Delay Function

The Twisting Triangle traversal is given by rotating a Reuleaux triangle (a triangle formed by the intersection of three circles placed on the corners of an equilateral unit triangle as depicted in Figure 5) counter-clockwise around the forwarder. The first node which is hit by this triangle becomes the next hop and repeats the process. The delay function specifically represents a rotating curve in form of the edge arc of a Reuleaux triangle with diameter r. Given forwarder v and its previous hop u, a candidate node w evaluates the delay function based on $d = |vw|$ and $\theta = \angle uvw$. The start angle depends on the distance to the previous

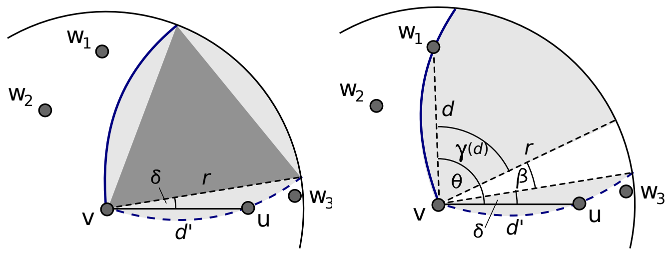


Figure 5: *Twisting Triangle starting at current node v with u as previous hop and δ being the start angle. The relative position of candidate w_1 is given by distance d and angle θ.*

hop $d' = |uv|$ and is given by $\delta = \arccos(d'/2r) - \pi/3$. The angle between the edge vw and the opposite edge of the enclosed equilateral triangle is given by $\gamma(d) := \arccos(d/2r)$ (see Fig. 5).

Using these preliminaries, a candidate calculates the delay by evaluating the following function of d and θ:

$$t(d,\theta) = \begin{cases} \dfrac{\theta - \gamma(d) - \delta}{5\pi/3} t_{\max} & \text{if } \theta - \gamma(d) > \delta \\ \infty & \text{otherwise.} \end{cases}$$

A delay of ∞ means that the candidate does not reply. The function is normalized by $5\pi/3$ and returns a fraction of $t_{\max}$, which is the maximum length of the contention period, an arbitrary fixed parameter. A denominator of $5\pi/3$ is chosen, because the Twisting Triangle does not perform a full turn, but always leaves out a sector of at least 60° between the previous hop and the first candidate (the start position of the triangle). We will later see that nodes in this area do not affect a correct boundary construction.

3.3 Combined RS Routing

When using the Sweep Circle as a rotating curve (and the corresponding delay function), it is shown in [24] that the RS algorithm traverses the α-shape. In conjunction with a greedy strategy, it forms the combined RS routing algorithm, which is a combination of contention-based greedy forwarding and the RS algorithm as a recovery strategy. It uses greedy forwarding whenever possible and starts the RS algorithm when reaching a local minimum. Then the RS algorithm traverses the boundary around a void region until greedy forwarding can be resumed.

Whenever the RS algorithm is started by a local minimum node v, the line $\overline{vT}$ toward the target is chosen as start angle (Figure 6). When applied by subsequent nodes, the RS algorithm yields a traversal path around a void region. The traversal path of the Twisting Triangle scheme and the Sweep Circle scheme (α-shape) are partially disjoint. In many cases it yields shorter paths as shown in Figure 4. In contrast to the Sweep Circle traversal the Twisting Triangle traversal depends on the start node and on the traversal direction. When started from a local minimum node it will produce a valid boundary traversal that guarantees progress in recovery.

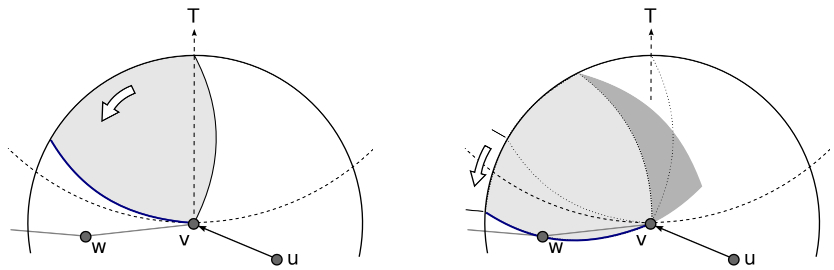


Figure 6: *Initialization of the RS algorithm at local minimum node v with start position (left) and forward triangle of vw (right) with the dark shaded critical region.*

3.4 Distributed Boundary Detection

Unlike in geographic routing, where a greedy failure marks the starting point of a boundary traversal, the distributed boundary detection requires that a node can identify by itself whether it is located on the boundary. A simple approach is to let each node wait for a random time after network initialization and then request a response of all neighbors. From the neighbors position, it checks the criterion of a (general) local minimum or stuck node. This involves all neighbors in the message exchange and requires a collision resolution. This can result in a significant communication overhead in dense networks, especially for those nodes who are far from the border and surrounded by many neighbors. By using the following contention-based algorithm using the idea of the sweep curve, we are able to limit the overhead to a constant number of messages. Furthermore, in the contention process, many collisions can already be avoided.

The following definition and preliminaries are used in the algorithm description: Given two nodes v and w, we denote the intersection of the circle C_1 having v as center and the circle C_2 having v and w on its perimeter as the lens $L_v(w)$. To be well-defined, both circles have the standard radius r, and for the center p of C_2, w is located left of vp (in counter-clockwise order), see Figure 7.

To describe a certain position of the sweep curve, we use a reference angle α, which is indicated by a dotted line in Fig. 7. For a point or node location on the sweep curve and inside v's radius given by angle θ and distance d, the reference angle is defined as $\alpha := \theta - \gamma(d) + \pi/2$ where $\gamma(d) := \arccos(d/2r)$. The following algorithm requires defined start and stop angles, which are used in the following delay function.

$$t(d, \theta, [\alpha_{\text{start}}, \alpha_{\text{stop}}]) = \begin{cases} \dfrac{\theta - \gamma(d) + \pi/2}{\alpha_{\text{stop}} - \alpha_{\text{start}}} t_{\max} & \text{if } \alpha_{\text{start}} \leq \theta - \gamma(d) + \pi/2 \leq \alpha_{\text{stop}} \\ \infty & \text{otherwise.} \end{cases}$$

The algorithm works as follows (cf. Figure [**?**]):

Phase 1: After chosing a random angle α_0, the initiating node v sends a FIND request. All neighbors use the aforementioned delay function to determine the time of their response. This is equivalent to a counter-clockwise rotation of the sweep curve until hitting the first

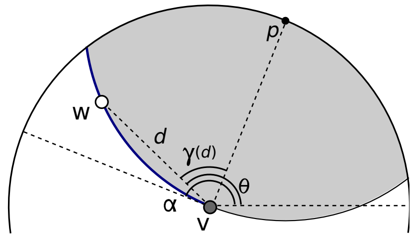


Figure 7: *Definition of the lens-shaped area $L_v(w)$ (shaded) and the reference angle α.*

node. This node would be the first to respond. If no node responds, v is a singular node. Otherwise, i.e. upon response of the first node w_1 (at angle ω_1), whose timer elapsed, the process is stopped immediately by the initiator v by sending a CHECK message. The check message initiates a sweep over the lens-shaped area $L_v(w)$. While sweeping over this area, the same delay function is used, but with a start angle opposite to ω_1. Only nodes in $L_v(w)$ area are allowed to answer. If there is no response, continue with phase 2a, if there is a response, continue with phase 2b.

Phase 2: The following steps are repeated until the angle of the first node w_1 is reached.

a) Initiator v issues a FIND message starting in clockwise order from α_i, the angle at which the check started. If there is no response, then the process stops. If there is a response, continue with 2b.

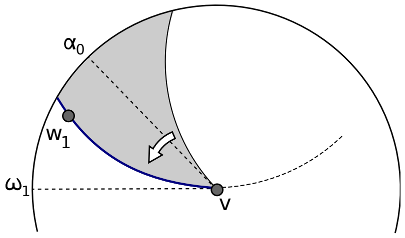
b) Initiator v issues a CHECK message starting in counter-clockwise order from α_i, which is opposite to ω_i. Here, the sweep curve changes orientation and is again restricted to $L_v(w_1)$. If there is no response, then we continue with phase 2a, finding further nodes. If there is a response, we continue with phase 2b.

As a result of the algorithm, we can analyze the distances between neighbors and identify void regions: If for two nodes $L_v(w_i)$ contains any other node w_j, otherwise we have found a local minimum.

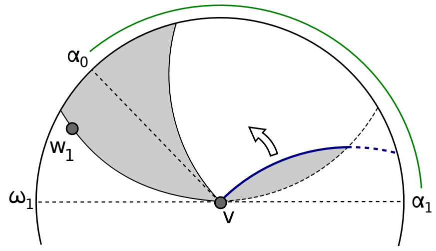
The neighborhood is discovered with at most 10 messages: After the Find-message by v, the first node w_1 (located close to the start angle) responds. Node v issues a check message, followed by a response by w_2 (located close to the first node). A Check-message follows, in turn responded by w_3. This check covers the area outside $L_v(w_1)$, thus $\angle w_1 v w_3 > 2\pi/3$. The following neighbors must have an angular distance of at least $2\pi/3$ to the second to last neighbor. Thus the maximum number of neighbors involved is 5, and at most 5 messages are sent by the forwarder.

4 Correctness of the Twisting Triangle Scheme

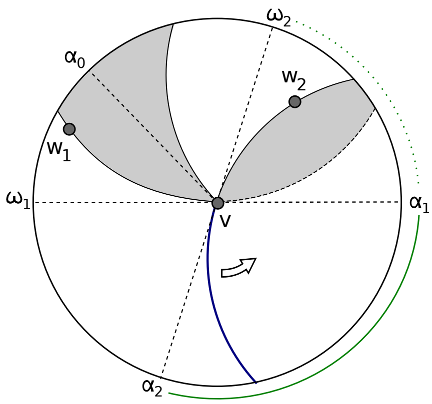
To prove the correctness of the Twisting Triangle scheme, we show that the traversal indeed produces a closed curve that encloses a void region, and that no nodes inside the void region are connected to nodes outside. Intersections and self-intersections do not lead to loops, nor



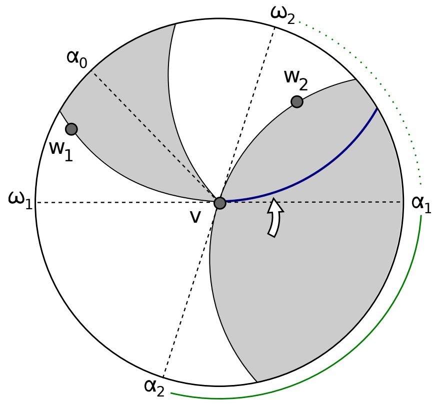
(a) FIND: Node v starts a sweep from a random angle α_0 until w_1 responds. The sweep stops at ω_1.



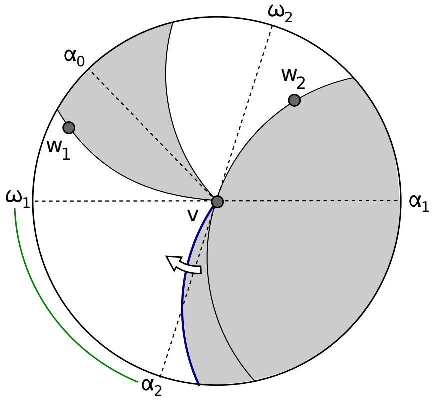
(b) CHECK: From the opposite angle α_1 a sweep is started until α_0 is reached or a second neighbor w_2 responds. Here, only nodes inside $L_v(w_1)$ may respond.



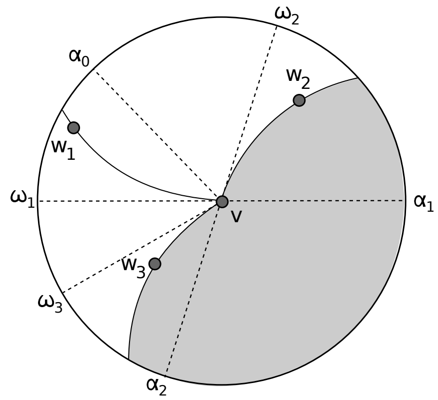
(c) CHECK: After the response of w_2 at angle ω_2, the sweep is started at the opposite angle α_2. It runs until α_1 is reached or the next neighbor responds.



(d) The previous angle α_1 is reached by the sweep without any response. The swept-over area is now large enough to fulfill the local minimum criterion.



(e) FIND: The area is enlarged by starting a sweep from α_2.



(f) Finally, w_3 responds. Since the angular distance to the first neighbor w_1 is too small, the process ends, revealing one local minimum in the direction of $w_2 v w_3$.

Figure 8: *Contention-based local minimum detection*

do they connect nodes outside the boundary with nodes inside the boundary. When started at a local minimum in georouting, the Twisiting Triangle traversal leads to a node which states a progress towards the target; in combination with a greedy strategy, which also provides a progress in each step, we can conclude that the target is eventually reached, whenever the network is connected.

Existing proofs on delivery guarantees of localized georouting algorithms rely on certain structural properties of planar subgraphs as described in the comprehensive analysis in [13]. These properties are not given here. Furthermore, the Twisting Triangle traversal is neither a subsequence of the α-shape for any real α, nor the boundary of the BoundHole algorithm [7]. Therefore, we can only follow the general concept of these existing proofs, but not apply them directly to this scheme. As in existing proofs we use the standard unit disk graph model and assume for convenience that no three nodes are co-circular.

4.1 Definitions and preliminaries

Definition 1. *A node v is called a* (general) local minimum*, if for every neighbor w of v holds $|vp| < |wp|$ and p is not in the transmission range of v (cf. the definition of a strongly stuck node in [7]).*

Definition 2. *We call the sequence of nodes that results by applying the RS algorithm started at local minimum node v a* traversal path *w.r.t. v. If not stated otherwise, the traversal is performed according to the right-hand rule, i.e. the next hop is found in counter-clockwise direction. Nodes on the traversal path are called* traversal nodes*, links between subsequent traversal nodes are called* traversal edges*.*

Definition 3. *A closed traversal path that is free of self-intersections and separates the plane into two regions, interior and exterior, where all nodes of a connected component are in the interior, is called a* valid boundary*.*

Definition 4. *We denote a Reuleaux triangle of width r and hinged at node v (i.e. having node v on one of its corners) as* backward triangle *or $\overline{N}_{TT}(wv)$, if node u is on its right border, and as* forward triangle *or $N_{TT}(vw)$ if u is on its left border. (see Figure 10). The forward triangle contains the* forbidden region *$N^{+}_{TT}(vw)$, which is left of vw, i.e. ahead of the sweep curve in counter-clockwise order (see Fig. 9).*

Definition 5. *We call the difference $N_{TT}(vu) \setminus \overline{N}_{TT}(uv)$, the* critical region *of the (directed) edge uv (see Fig. 16).*

4.2 Correctness Proof

The fist two lemmas show that the traversal is accompanied by a sequence of empty forward triangles.

Lemma 1. *Let v be a local minimum node. Then the critical region of vw is empty.*

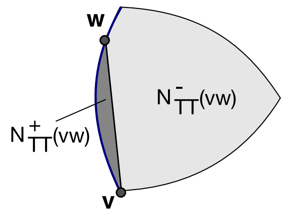


Figure 9: *Definition of* $N^{+}_{TT}(vw)$ *and* $N^{-}_{TT}(vw)$

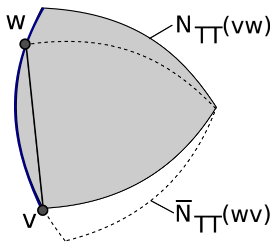


Figure 10: *Forward triangle* N_{TT} *and backward triangle* $\overline{N}_{\mathrm{TT}}$

Proof. The empty region of a local minimum node v (i.e. the part of the transmission area, in which no other node is closer to the target) is minimized, if the target T is located near the boundary of the transmission area of v, i.e. |vt| = $r+\varepsilon$ and for any neighbor w of v holds |wt| $> r+\varepsilon$. When starting the rotational sweep, the Reuleaux triangle is inside the empty region (see Fig. 6) because the width of the triangle does not exceed r. For the triangle in this position holds that for any point p on the sweep curve (left bold edge of the triangle in the figure) holds that also the backward triangle $\overline{N}_{\mathrm{TT}}(pv)$ is in the empty region, which contains the critical region. After starting the rotational sweep, the critical region remains empty: Assume the contrary, that there is a node x in the critical region of link vw. Both nodes w and x must be outside the empty region of the local minimum v. For a sweep in counter-clockwise order holds $\angle tvw > \angle tvx + \pi/3$ because sweep line and critical region are separated by at least $\pi/3$. From the smaller angle follows that x is then selected as next neighbor instead of w, which is a contradiction. In other words, if a node was inside the critical region, it would have been hit by the sweep line before. □

Lemma 2. *For any edge vw of the traversal path holds that the forward triangle* $N_{TT}(vw)$ *is empty.*

Proof. Assume the contrary, there is a node inside $N_{\mathrm{TT}}(vw)$. The predecessor u of v must be located outside $N_{\mathrm{TT}}(vw)$, otherwise vw was not a traversal edge. Though u must be outside the forward triangle, u's predecessor x could be inside. Here, only the intersection of the forward triangle (shaded lightly) and the critical region of uv (shaded dark) is a possible location. But this part is fully covered by the circle $C(v,r)$, thus x would be the predecessor of w and not of u, i.e. uv would not be a traversal edge. □

The following lemmas are used to prove that a traversal path does not contain unwanted intersections at nodes and edges.

Lemma 3. *If a node v appears more than once on a traversal path as uvw and* $u'vw'$*, then the angles* $\angle uvw$ *and* $\angle u'vw'$ *are disjoint.*

Proof. Assume v appears twice on a traversal path with different predecessors u and u'. Furthermore assume the contrary, that the angles are indeed overlapping, i.e. $\angle uvu' < \angle u'vw'$. By

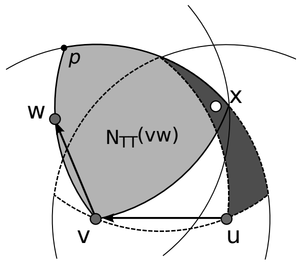


Figure 11: *Illustration for Lemma 2*

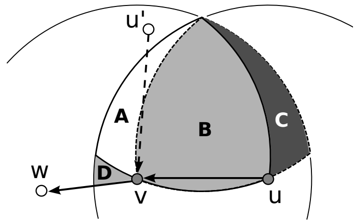


Figure 12: *Illustration for Lemma 3*

Lemma 2 the forward triangle $N_{\mathrm{TT}}(uv)$ (union of A and B in Fig. 12) is empty, thus u' must be located outside this region. If $\angle uvu' > \pi/3$ then u' becomes the successor of v and uvw is not part of the traversal path. If $\angle uvu' < \pi/3$, then the edge $u'v$ can only exist if u' is located in the critical region of uv (see Fig. 12). Then u' must have been selected by a predecessor y' that is not u's predecessor. Otherwise yu' and $u'v$ are traversal edges, but not uv. This predecessor y' is located in the critical region of yu' (containing locations that prevent the edge yu'). This prerequisite holds recursively, as long as there is a node in the corresponding critical region. Since we require a traversal path to start with a local minimum, where the critical region is empty, this finally leads to a contradiction. ☐

Lemma 4. *If an edge uv is part of a traversal path P, then there is no edge wx intersecting uv where x is neither connected to u nor v.*

Proof. If uv is a traversal edge, there is an empty Reuleaux triangle $N_{\mathrm{TT}}(uv)$ hinged at u and with v on one of its sides. Let p be the corner opposite to this side, which is by definition located at a distance of r from u and also from v (see Figure 13). Assume that there is an edge wx intersecting uv. Then there is a node w left of uv and a node x right of vw. Node w must be outside $N^+_{\mathrm{TT}}(uv)$ – otherwise uv is not a traversal edge. For simplicity we move w on the border of $N^+_{\mathrm{TT}}(uv)$, which can only reduce the distance to x. Assume, without loss of generality, that x and u are below wp. Any node connected to w (including x) must be inside the circle $C(w,r)$, i.e. the transmission range of w (solid circle in Figure 13). This circle can be obtained by rotating $C(u,r)$ hinged at p such that u moves towards w. The part of this circle that is right of uv and below wp remains entirely inside $C(u,r)$. Therefore x is also inside that circle, and subsequently $ux \leq r$. It means that x is connected to u, which is a contradiction. ☐

Finally, the next lemma leads to the proof that the traversal is a simple closed curve.

Lemma 5. *Let* $P := (v_0, v_1, \ldots, v_n)$ *be a traversal path with* v_0 *being a local minimum node. Then the following holds:*

a) The path is not self-intersecting.

b) The path is finite and closed, i.e. $v_0 = v_n$.

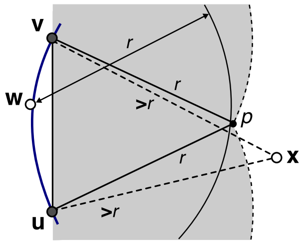


Figure 13: *Illustration for Lemma 4*

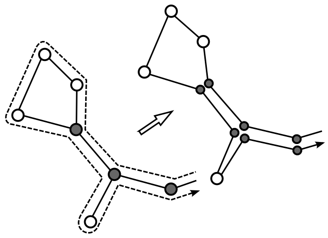


Figure 14: *Splitting re-visited nodes (see Corollary 1)*

c) The path P encloses all nodes of a connected component.

Proof. a) Assume $v'w'$ in Fig. 17 is a traversal edge, which intersects another traversal edge vw. v' is located in the critical region of vw' so that vw' does not become part of the traversal path. v' itself can only be a traversal node if it has a predecessor different than u, which is in turn located in the critical region of uv'. This construction can be followed recursively as long as there are nodes in the critical region of boundary nodes that are adjacent in the unit disk graph (see Fig. 15 for an example). As we require that the start node v_0 is a local minimum node, the critical region of any edge starting from v_0 is empty, which is a contradiction.

b) From Lemma 3 and the fact that previous hop and next hop are at least separated by an angle of $\pi/3$ (Lemma 2), it follows that a node cannot appear infinitely often on a traversal path, thus the path is finite and connects start node v_0 and end node v_n.

c) Assume the contrary, i.e. there is a non-traversal edge from the exterior to the interior of the face. Then there must be a traversal edge vw that is intersected by an edge $v'w'$ as depicted in Fig. 17. By Lemma 4 v' is connected to at least one of the nodes v and w. Assume w.l.o.g. that v' is connected to v as in the figure. As $v'w'$ is a non-traversal edge, v is not the successor of v', which only holds if there is another node u in the critical region of $v'v$ (dark shaded area in the figure). Then v' is left of uv and not in the interior of the enclosed face. □

Corollary 1. *The TT traversal path is a simple closed curve.*

Proof. From Lemma 2 we know that for a subsequence uvw of the traversal path the angle $\angle uvw$ in traversal direction is at least $\pi/3$, because a sector of $\pi/3$ is contained in the empty forward triangle. With Lemma 3, it follows that a node cannot appear infinitely often on a traversal path. As we can virtually split the re-visited nodes into multiple nodes as in Fig. 14 (cf. Theorem 3.11 in [8]), we obtain at a simple closed curve. □

4.3 Guaranteed Delivery in Georouting

As a consequence of the previous proofs that show that the Twisting Triangle scheme produces a valid boundary along void regions, we can conclude that it provides the delivery guarantee to geographic routing algorithms (here: combined RS routing). The same arguments as in [24] apply and are ommitted here due to space limitations.

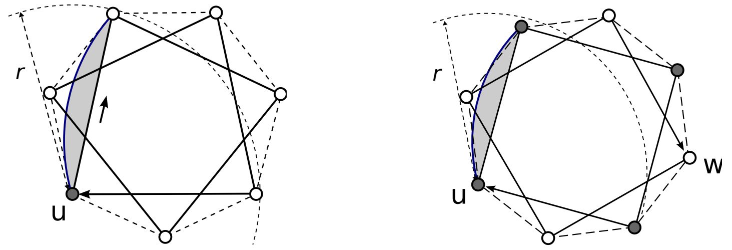


Figure 15: *Self-overlapping traversal path (left) and intersecting disjoint paths (right). Both have no local minimum node.*

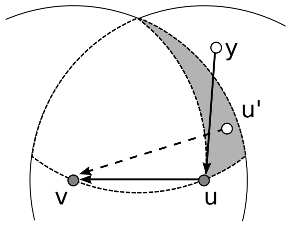


Figure 16: *Critical region of edge uv*

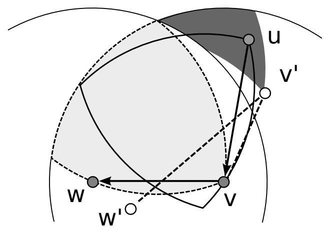


Figure 17: *Illustration for Lemma 5*

Lemma 6. *Let v and T be two nodes in a network graph that can reach each other and v a local minimum node. When starting the RS algorithm at v, the TT traversal path leads to a node v' with $|v'T| < |vT|$.*

Proof. A proof is given in [24]. It follows from the fact that a void region can be enclosed by a traversal which does not allow any paths leading to the interior (Lemma 5). □

Theorem 1. *Greedy forwarding combined with Rotational Sweep using the Twisting Triangle scheme guarantees delivery.*

Proof. This follows from the fact that greedy forwarding as well as the traversal path generated by RS bring a packet closer to the destination (Lemma 6). A proof is given in [24]. □

5 Properties of the Twisting Triangle scheme

5.1 Locally Optimal Neighbor Selection

In localized routing and boundary traversal, it is not possible to select the globally optimal next hop. Choosing the largest advance in one step can lead to smaller advancements in the next step as the example in Figure 20 shows. In a boundary traversal the next edge in counter-clockwise order is a natural choice. The BoundHole algorithm selects the next neighbor in counter-clockwise order directly (while forbidding backward nodes). However, intersections

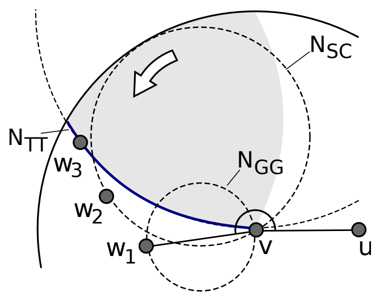


Figure 18: *Comparison of next hop identification criteria*

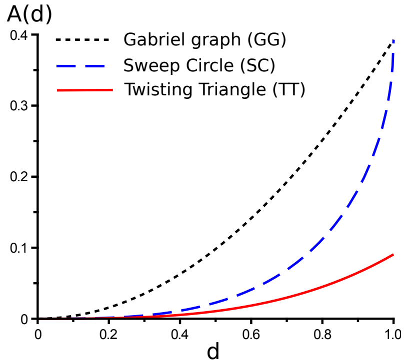


Figure 19: *Area of the forbidden region $N^+(vw)$ based on the distance $d = |vw|$.*

of this edge might occur, which cannot be detected locally by the forwarder nor by the next neighbor (Fig. 3).

Reactive, localized algorithms, cannot resolve such situations. Therefore, Gabriel subgraph constructions, Sweep circle, and Twisting Triangle schemes avoid such edges by preventively excluding them. An neighbor selection is only valid if the forbidden region of the respective edge is empty (see Fig. 18). If not, the edge is replaced by an intermediate node. This replacement leads to shorter edges and increased routing or traversal paths. Hence, the smaller the forbidden region, the more edges can be preserved locally. The likelihood of preserving an edge is given by the probability of an empty forbidden region: $\mathrm{P}[X = 0] = e^{-\lambda A(d)}$, where λ is the node density, d the distance to the neighbor, $A(d)$ the corresponding area. Fig. 19 shows this area (independent of the node density) for different schemes. It is minimal for the Twisting Triangle scheme

Lemma 7. *The Twisting Triangle minimizes the area of the forbidden region $N^+(uw)$ such that uw is free of any intersecting edge vx where one endpoint x is neither connected to u nor v.*

Proof. The forbidden region of the Twisting Triangle scheme is by definition the circular segment of a unit disk circle centered at a point p which is equidistant to u and v (see Fig. 13). Assume the forbidden region of any traversal scheme is smaller than $N^+_{\mathrm{TT}}(uw)$. Then a third node v is allowed inside $N^+_{\mathrm{TT}}(uw)$ which is not chosen as successor of u and which can be connected to another node x that is neither connected to u nor to w as shown in Fig. 3. $\square$

This implies that the next neighbor selection is locally optimal, as it combines the smallest forbidden region, which preserves edges, with the property that no important links are missed and routing loops can be avoided.

Algorithm	Min. Delay	Max. Delay
BLR Request-Response [18]	$t_{\max}$	$t_{\max}$
Angular Relaying (AR) [20],	$1/4\,t_{\max}$	$t_{\max}$
RS, Sweep Circle [24],	0	$t_{\max}$
New RS, Twisting Triangle (this work)	0	$5/6\,t_{\max}$

Table 3: *Overview of worst-case delays by contention-based traversal algorithms*

5.2 Worst-Case Delay

In contention-based routing the per-hop delay is determined by the delay function and the location of a neighbor. If such neighbor does not exist, the forwarder has to wait a maximum time span until the message is returned to the previous node. In comparison to previous approaches the Twisting Triangle scheme has a shorter worst case delay (see Table 3), because the sweep curve does not perform a full rotation in the worst case. We can decrease $t_{\max}$ by $5/6$ without increasing the rotation speed of the sweep curve itself

5.3 Relation to α-shapes and Planar subgraph traversals

The Sweep circle scheme yields an α-shape with $\overline{\alpha} = \frac{r}{2}$ which is a subsequence of a Gabriel subgraph boundary as shown in [24]. The Twisting Triangle scheme is partially dijoint to the α-shape boundary and to the Gabriel subgraph boundary. Figures 20 and 21 show counter-examples. However, it is a subsequence of the relative neighborhood (RNG, [19]) subgraph boundary.

Theorem 2. *Let S_{GG} and S_{RNG} be sequences of nodes by a face traversal on a Gabriel subgraph and on a RNG subgraph. Let S_{SC}, and S_{TT} be the sequence of nodes obtained by the Sweep Circle scheme and the Twisting Triangle scheme. Then it holds that*

1. $S_{\mathrm{RNG}} \subseteq S_{\mathrm{GG}} \subseteq S_{\mathrm{SC}}$
2. $S_{\mathrm{GG}} \not\subseteq S_{\mathrm{TT}}$ *and* $S_{\mathrm{SC}} \not\subseteq S_{\mathrm{TT}}$
3. $S_{\mathrm{RNG}} \subseteq S_{\mathrm{TT}}$

Proof.

1. The subset relation of $S_{\mathrm{RNG}} \subseteq S_{\mathrm{GG}}$ follows from the fact that the Gabriel graph is a subgraph of the RNG. The subset relation of $S_{\mathrm{GG}} \subseteq S_{\mathrm{SC}}$ is shown in [24].
2. Figures 20 and 21 give examples where nodes are part of S_{GG} or S_{SC}, but not part of S_{TT} and vice versa.
3. Assume the contrary, that there is an edge $xw \in S_{\mathrm{RNG}}$ which is not part of S_{TT} and intersects and edge $uv \in S_{\mathrm{TT}}$. Then this edge must span the triangle as depicted in Fig. 22. In this setting, it is not possible that the endpoint v is not contained in the RNG lens over wx, which has to be empty.

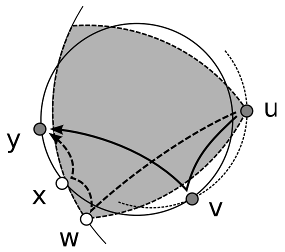


Figure 20: *TT traversal (uwxy) and α-shape traversal (uvy) are partially disjoint*

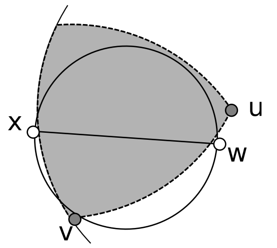


Figure 21: *TT traversal (uvx) and Gabriel graph traversal (uwx) are partially disjoint*

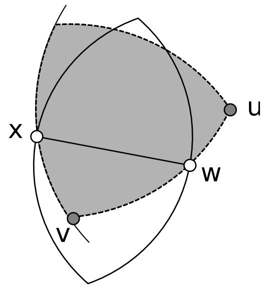


Figure 22: *There are no RNG edges intersecting a Twisting Triangle traversal*

The inequality, i.e. $S_{\mathrm{TT}} \not\subset S_{\mathrm{RNG}}$ follows from the observation that for any edge uv the area $N^{+}_{\mathrm{TT}}(uv)$ is a strict subarea of the RNG lens. Thus a node might be inside the RNG lens over uv, which prohibits this edge, but not contained in $N^{+}_{\mathrm{TT}}(uv)$.

□

Note, that not every worst case example for the RNG is also a worst case for a TT traversal.

6 Simulations

We have performed simulations on random networks comparing the new RS algorithm using the Twisting Triangle scheme with the following algorithms: (1) BLR Request-Response [18] triggers replies from all neighbors and constructs the Gabriel subgraph. (2) Angular Relaying (AR) [20] selects the first node in counter-clockwise order as candidate for the Gabriel subgraph and allows other nodes to protest and correct the decision. The result is a traversal of the Gabriel subgraph boundary. (3) The RS algorithm with the Sweep Circle scheme [24] yields traversals of the α-shape.

Simulation Setup

We performed simulations with our own custom simulator using a simplified MAC layer model, which assumes uniform transmission radii and no collisions. In each simulation run nodes are placed randomly and uniformly in the simulation area. The transmission range is fixed and determined such that the resulting network has a certain density (average number of neighbors). A circular void region covering 20% of the simulation area is optional. After selecting a local minimum node as start node, one message is routed around the boundary until returning the start node again. We record the number of hops and 'net' delay by the contention process for each hop. For the delay statistics the time for sending control messages and the inter-frame spaces are excluded, as they depend on the physical/MAC layer implementation of the target platform and not depending on the algorithm. Simulations were performed on

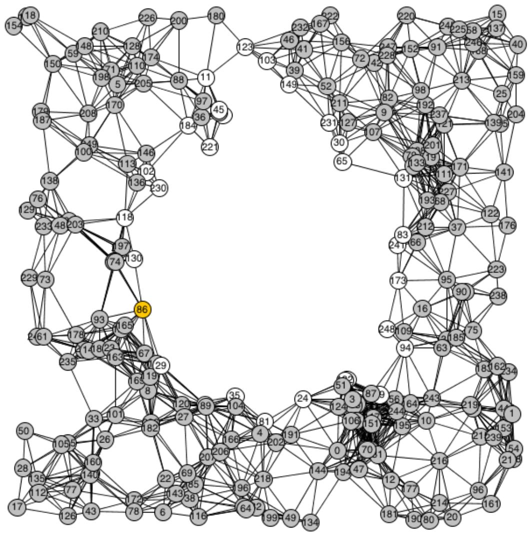


Figure 23: *Unit disk graph with density 6 and circular void region in the center. Inner boundary nodes are white, other nodes are shaded.*

100 random node sets with 250 nodes each, and for different network densities ranging from 4 to 14. Non-connected networks were excluded.

Simulation Results

Boundary Size: Comparison of boundary lengths are shown in Figure 24 (left). In this respect the Twisting Triangle scheme provides the shortest boundaries among all reactive algorithms. In comparison to the Sweep Circle traversal (the best known algorithm), boundaries are shorter by up to 20% on average for higher network densities on outer boundary traversals and up to 30% on average for inner void regions. The reason is that the Twisting Triangle scheme has the smallest forbidden region (cf. Fig. 19), which results locally in longer edges and globally in shorter boundaries.

Per-node delay: Figure 24 (left) show the 'net delay' per node, which is the length of the contention period between reception of the RTS and transmission of the CTS by the first candidate. In this respect, the Sweep Circle traversal provides the shortest contention delays, which are up to 27% shorter than the Twisting Triangle traversal. This delay does not include the time for message transmissions and inter-frame spaces. We denote this additional time overhead by t_{add} and include it in the following considerations on end-to-end delay.

End-to-end delay or latency: The end-to-end delay D is the sum of the delays t_i by the contention period of the i-th node on the traversal path and the additional overhead t_{add} per hop: $D = \sum_{i=0}^{L} t_i + t_{\text{add}}$. The overhead t_{add} depends on the physical/MAC layer of the target platform and the choice of $t_{\max}$. Instead of making assumptions about $t_{\max}$, which could favor one of the approaches, we calculate the break even point, i.e. the overhead t'_{add} where the

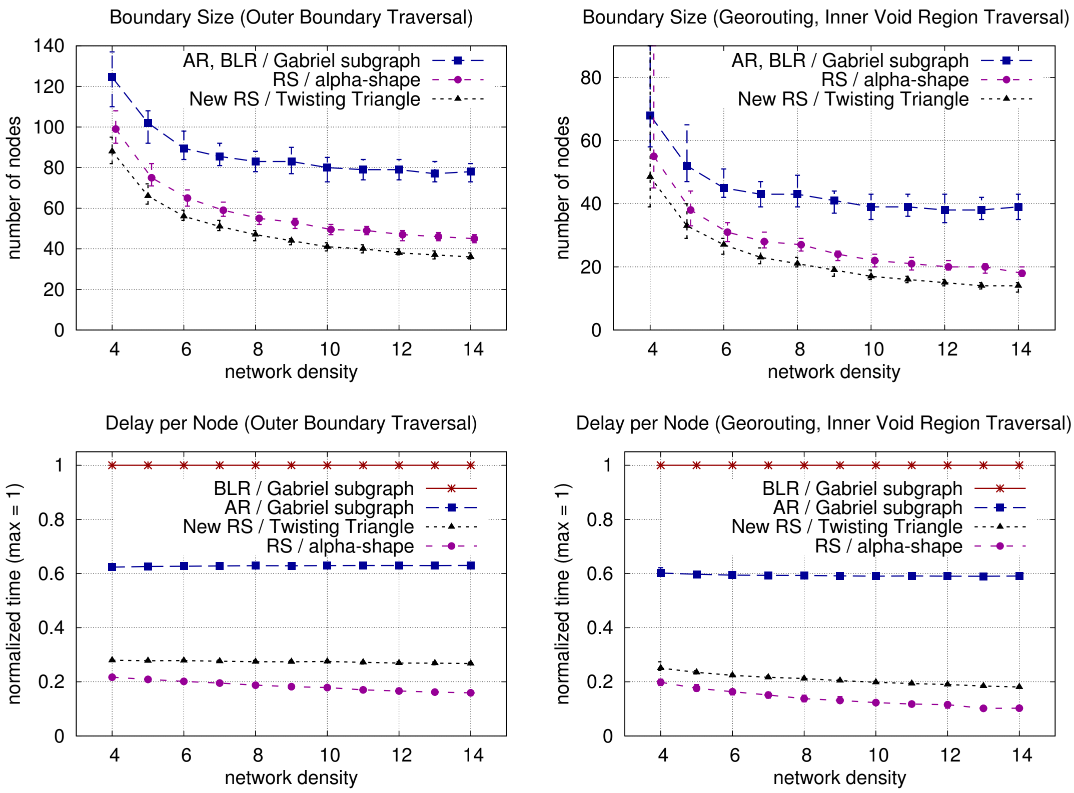


Figure 24: *Comparison of boundary size and per-node delays for different algorithms and the respective subgraph constructions. The plots show median and quartiles of 100 random graphs.*

Sweep Circle traversal and the Twisting Triangle traversal have equal end-to-end delays (L is the length of the traversal): $t'_o = (D_{TT} - D_{SC})/(L_{SC} - L_{TT})$

In simulations the break even point is $t'_{\text{add}} \approx 0.3t_{\max}$ four outer boundary traversals and ranging from $0.12t_{\max}$ to $0.24t_{\max}$ for inner void region traversals. Hence, for the simulated random networks, if the time overhead for message transmissions including inter-frame spaces is greater than $0.3t_{\max}$, then Twisting Triangle traversal can be performed faster than the Sweep Circle traversal.

7 Conclusion

We have presented a new fully-reactive traversal algorithm for georouting and boundary detection, that does not need prior neighborhood knowledge. It preserves as much edges as possible – in contrast to existing approaches – and avoids critical edges that might have locally undetectable intersections, which can lead to routing loops. In this respect the algorithm makes locally optimal decisions. Globally, it provides shorter boundaries on average than existing approaches for constructing planar subgraphs or planar boundaries. The reason is that planarization is not a prerequisite for correctness or delivery guarantees. The new Twisting Triangle scheme can be used in fully-reactive contention-based routing algorithms or in conventional routing as well.

References

[1] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," University of Virginia, USA, Tech. Rep. CS-2003-11, Apr. 2003.

[2] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, "On the spanning ratio of gabriel graphs and beta-skeletons," *SIAM Journal on Discrete Mathematics*, vol. 20 (2), pp. 412–427, 2006.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. 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)*. ACM Press, 1999, pp. 48–55.

[4] D. Chen and P. K. Varshney, "A survey of void handling techniques for geographic routing in wireless networks," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 1, pp. 50–67, 2007.

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

[6] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. IT-29(4), pp. 551–559, 1983.

[7] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *23rd Annual IEEE Conference on Computer Communications (INFOCOM)*, March 2004, pp. 2458–2468.

[8] ——, "Locating and bypassing routing holes in sensor networks," *Mobile Networks and Applications*, vol. 11, pp. 187–200, 2006.

[9] M. Fayed and H. T. Mouftah, "Localised alpha-shape computations for boundary recognition in sensor networks," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1259–1269, 2009.

[10] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," University of Southern California, Tech. Rep. ISI/RR-87-180, Mar. 1987.

[11] H. Frey and S. Rührup, "Paving the way towards reactive planar spanner construction in wireless networks," in *GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS 2009)*, Mar. 2009.

[12] H. Frey, S. Rührup, and I. Stojmenovic, "Routing in wireless sensor networks," in *Guide to Wireless Sensor Networks*, S. Misra, I. Woungang, and S. C. Misra, Eds. Springer, London, May 2009.

[13] H. Frey and I. 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.

[14] H. Füßler, J. Widmer, M. Mauve, and H. Hartenstein, "A novel forwarding paradigm for position-based routing (with implicit addressing)," in *IEEE 18th Annual Workshop on Computer Communications (CCW 2003)*, 2003, pp. 194–200.

[15] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18 (3), pp. 259–278, 1969.

[16] J. Gao, L. J. Guibas, J. E. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *2nd Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)*, Oct. 2001, pp. 45–55.

[17] M. Heissenbüttel and T. 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*, 2003, pp. 197–209.

[18] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "BLR: Beacon-less routing algorithm for mobile ad-hoc networks," *Computer Communications*, vol. 27, no. 11, pp. 1076–1086, Jul. 2004.

[19] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proc. of the IEEE*, vol. 80, pp. 1502–1517, 1992.

[20] H. Kalosha, A. Nayak, S. Rührup, and I. Stojmenovic, "Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks," in *27th Annual IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2008.

[21] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *4th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, 2003, pp. 267–278.

[22] 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.

[23] W.-J. Liu and K.-T. Feng, "Greedy anti-void routing protocol for wireless sensor networks," *IEEE Comm. Letters*, vol. 11(7), pp. 562–564, 2007.

[24] S. Rührup and I. Stojmenovic, "Contention-based georouting with guaranteed delivery, minimal communication overhead, and shorter paths in wireless sensor networks," in *24th Int. Parallel and Distributed Processing Symposium (IPDPS 2010)*, April 2010.