Prohibitive-link Detection and Routing Protocol

Marwan Fayed and Hussein T. Mouftah[*]
School of Information Technology & Engineering
University of Ottawa

TR-2008-04

Abstract

In this paper we investigate the limits of routing according to left- or right-hand rule (LHR). Using LHR, a node upon receipt of a message will forward to the neighbour that sits next in counter-clockwise order in the network graph. When used to recover from greedy routing failures, LHR guarantees success if implemented over planar graphs. We note, however, that if planarity is violated then LHR is only guaranteed to eventually return to the point of origin. Our work seeks to understand why. An enumeration and analysis of possible intersections leads us to propose the Prohibitive-link Detection and Routing Protocol (PDRP) that can guarantee delivery over non-planar graphs. As the name implies, the protocol detects and circumvents the 'bad' links that hamper LHR. Our implementation of PDRP in TinyOS reveals the same level of service as face-routing protocols despite preserving most intersecting links in the network.

1 Introduction

The construction of network subgraphs appropriate for position-based (or geographic) routing protocols has, to date, remained a complex problem. These subgraphs are needed to recover from the local minima problem (see [3]) that prevents delivery and plagues position-based protocols. Network subgraphs constructed for recovery using only 1-hop information risk inaccuracies that cause

[*] Marwan Fayed and Hussein T. Mouftah are supported by the Natural Sciences and Engineering Research Council of Canada. Email: {mmf,mouftah}@site.uottawa.ca.

routing failures([16, 25]). If permitted to cooperate, nodes may construct a network subgraph that remedies any inaccuracies([15, 25]). Yet the energy needed to power many rounds of communication risks being prohibitive in such a resource-constrained environment. The ideal wireless network subgraph would guarantee successful delivery while a) needing only 1-hop information and b) be able to acquire such information passively.

Traditionally, position-based routing protocols construct subgraphs (herein referred to as just 'graph') from available links in somewhat of a bottom-up fashion. Generally the idea is to extract a specific type of graph. During the setup of the graph each node evaluates potential links to find those that preserve some global properties. Planar graphs([5]) and k-spanners([24]) are two such examples. The analogous question would be to ask, "what is the minimum set of edges that must be preserved to guarantee so-and-so feature in the graph?"

Our work is motivated by the opposite question, "What is the minimum set of edges that must be *deleted* while still providing guarantees?" Without sacrificing the scalability and success of position-based routing, the goal of this work is to disturb the network as little as possible. To this end it is necessary to understand the causes for a position-based routing protocol to fail to recover from local minima and deal with those causes, directly. We believe this paper is the first work in that direction.

In this paper we investigate routing according to left- or right-hand rule (LHR). Using LHR, a node upon receipt of a message will forward to the neighbour that sits next in counter-clockwise order in the network graph. (Alternatively, clockwise order if using right-hand rule.) When used to recover from greedy routing failures, LHR guarantees success if implemented over planar graphs; for this reason it is often called 'face-routing'. We note, however, that if planarity is violated then LHR is only guaranteed to eventually return to the point of origin. Our work seeks to understand and correct the underlying causes.

We have chosen LHR for three reasons. First, it is most prevalent in position-based routing literature and hence well-studied. Second, it is a simple rule. Finally, the ideal network graph remains elusive. To re-iterate, we envision the ideal graph as overcoming the inaccuracies that lead to routing failures; as one that results from knowledge of the 1-hop neighbourhood; as one where each node transmits a constant number of messages.

We begin by with a provable enumeration of the possible types of intersections in a unit-disc graph, within which any two nodes are neighbours if separated by a maximum distance of one unit. We show that only three types of intersections are possible and that in only one of these configurations does LHR fail to recover.

2 Related Work

The most prominent and best known recovery algorithms route around the hole face (or perimeter) in the planar subgraph. This method is equivalently known as *face routing* ([1, 2]) and *perimeter routing* ([13]). Face routing was first proposed by Bose et al. in [1] with some theoretical bounds. Karp et al. independently proposed an identical mechanism in [13] but with work on a MAC-compatible implementation. Variants have since emerged addressing, for example, theoretical bounds in [18, 17, 19]. In [11], face-routing is augmented into a "select-and-protest" reactive protocol in order to reduce the information required to planarize the graph.

Wireless network graphs may consist of intersecting edges so it is necessary for planar subgraph method to prune edges from the network graph so that it is planar while remaining connected. Gabriel Graphs (GG) and Relative Neighbourhood Graphs (RNG) are planar graphs whose constructions are localised, a characteristic particularly suitable to sensor environments. Intersecting edges are eliminated by connecting pairs of nodes through *witness* nodes, if such a node exists in a common region. It has since been shown that 'Hello' messages may hinder network performance [8]. This is addressed in face-routing directly by [4] and more generally in [6, 9, 23]. Further work in [27] reduces the path length during the recovery phase.

These distributed constructions are unable to resolve links broken by obstacles or interference [12, 22]. Recent breakthroughs have begun to surmount the impracticalities of face-routing while maintaining delivery guarantees [15, 20].

The Greedy Distributed Spanning Tree Protocol (GDSTR) algorithm in [20] builds on the fact that any message can be successfully delivered via depth-first search if the network is connected via a spanning tree. (This fact alone does not solve the problem: delivery would be inefficient, needing up to $2n - 3$ hops.) The authors in [20] describe a new type of spanning tree, the *hull tree*, to route more efficiently. A hull tree is a spanning tree with one added piece of information: each node records the convex hull that contains all of its descendants in the tree. (The convex hull of a set of points is the smallest polygon that contains all the points.) In GDSTR forwarding occurs greedily, as with most position-based protocols. If a message reaches a void, a recovery mode is initiated where convex hulls are used to determine the regions of the network that contain unreachable destination. This information is used by GDSTR to route along the spanning tree to forward to the appropriate convex hull. If a node is found en route that is closer to the destination than the node where the message was stuck, then GDSTR returns to greedy forwarding. GDSTR is known to scale well as the neighbourhood size grows. Furthermore, the use of multiple hull trees adds fault-tolerance to the network and if multiple trees are rooted at opposite ends of the network, routing efficiency improves.

The Cross-Link Detection Protocol (CLDP) proposed in [15], and later improved in [7], circumvents voids by face-routing. It uses left-hand rule over a planar subgraph of the network; its design however, is motivated by the observation that routing difficulties in planar subgraph methods arise, in part, due to the constructions themselves. (Recall from previous that successful local planar subgraph constructions rely on the unit disc graph.) For this reason, CLDP proposes an alternate construction of planar subgraphs that assumes only that links are bidirectional. CLDP operates in a distributed fashion, exchanging some localised operation for accurate information. The idea behind CLDP is that each node is able to probe the vicinity for intersecting links. A probe packet is initialised with the endpoints of the first link to be probed and forwarded according to left-hand rule. The probe eventually returns to its point of origin with a vector of the path taken. This information is shared with nearby nodes to prune links appropriately. To avoid the slow process of scheduling serial probes by neighbouring nodes, a system for concurrent probing is proposed. Concurrent probing is achieved by implementing a mechanism to 'lock' links so that no more than one link is removed at a time from any vicinity. CLDP is one of very few protocols to have been implemented on testbeds [15]. The associated communication complexities and storage costs revealed in this process (see [16, 20]) are motivation to develop alternative approaches to guarantee delivery.

Protocols such as CLDP and GDSTR, in order to be feasible for physical networks, sacrifice efficiency for accuracy. CLDP requires high-complexity negotiations within each neighbourhood in order to prune appropriate links, and GDSTR must broadcast information to construct and maintain its hull trees. It remains an open question whether such trade-offs are a necessity.

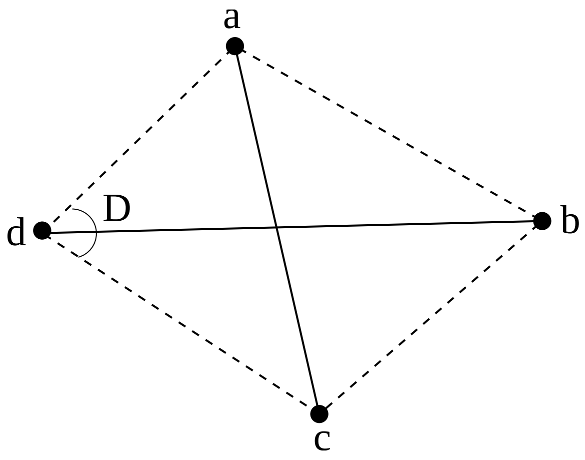


Figure 1: Intersecting links between two pairs of nodes may impose any or all edges in a 4-gon.

By contrast, in this work we show that there exists a locally constructed, non-planar graph construction over which face-based protocols guarantee success.

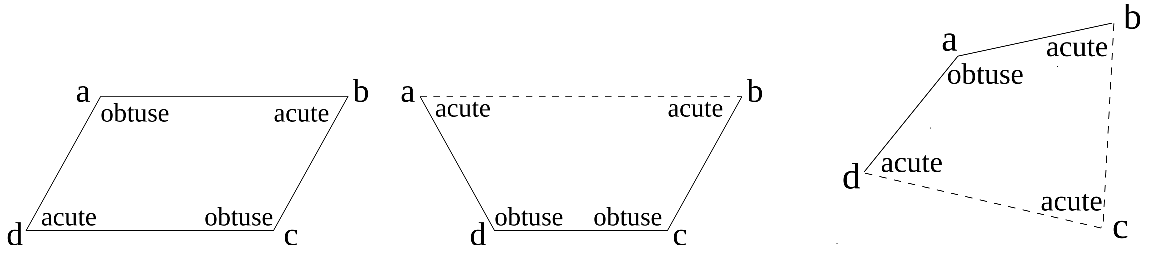
3 Links that Prohibit Routing Success

In the previous section we noted that routing according to right- or left-hand rule (LHR), alone, fails to provide a guarantee of success. Though this fact is well known, the reasons and circumstances under which delivery may fail are poorly understood. In this section we seek to investigate the limits imposed by intersections on face-based recovery.

Our goal is to maximise the number of active edges in a wireless network graph while providing routing guarantees using LHR. In an attempt to relax planarity, currently required to guarantee the success of a traversal between two nodes, we must identify the causes for failure in an arbitrary graph. We focus this work on the unit disc graph (UDG), where all communication ranges are normalised. The UDG is appropriate since it limits potential routing options yet still poses a challenge to LHR routing. Our investigation begins with an enumeration of all of the types of intersections that may appear in the UDG.

3.1 An Enumeration of Intersection Types

Consider any two intersecting edges. We provide the edges ac and bd in Figure 1 for reference. The nodes a, b, c, d at the end points of these edges form a 4-gon (shown in Figure 1 using dashed lines).



(a) All edges of the 4-gon must exist.

(b) Edges bc, cd, da must exist.

(c) Edges ab, ad must exist.

Figure 2: Possible 4-gons when edges ac and bd intersect in the UDG with dashed lines indicating edges that may or may not appear.

The question we ask is, which of the edges of the 4-gon may or may not be communicating links in the unit-disc graph? In order for at least one such edge to exist, we need to show that *all four sides cannot be greater than both diagonals.*

Using cosine rule we know,

$$(ac)^2 = (ad)^2 + (dc)^2 - 2(ad)(dc)\cos D. \tag{1}$$

If $|ac|$ is less than or equal to 1, then

$$(ad)^2 + (dc)^2 - (ad)(dc)\cos D \leq 1. \tag{2}$$

When $D \geq \frac{\pi}{2}$, then $\cos D \leq 0$. In this case, $(ad)^2 + (dc)^2 \leq 1$, which means $(ad) \leq 1$ and $(ad) \leq 1$. Thus, if an angle of the 4gon is right or obtuse, then both incident edges must exist in the UDG. (By contrast, incident edges when $D < \frac{\pi}{2}$ may or may not exist.)

This implies and restricts the possible configurations that allow intersections to three, all shown in Figure 2. The two cases where the nodes of intersecting edges produce a 4-gon with two obtuse angles is shown in Figures 2aand 2b, while the 4-gon containing a single obtuse angle is shown in Figure 2c. (Note that it is impossible for a 4-gon to be constructed with three obtuse edges; and that edges incident to an acute angle may or may not appear in the unit-disc graph.)

3.2 The Prohibitive Link

A trace over each configuration reveals that LHR routing recovers from all but one configuation. The 'bad' configuration occurs during a traversal of the intersection in Figure 2c where there are three acute angles and when only the edges of the 4-gon incident to the obtuse angle appear. This represents the case where network node a communicates with b, c, d, and b with d; node c communicates only with a. We call this case the *umbrella* configuration for its appearance, as shown in Figure 3.

Referring to Figure 3 there are two ways in which LHR may fail. The first is demonstrated by the dashed-dot-dash line originating at node d. (Entry at nodes a and b are analogous.) A traversal using left- or right-hand rule will never traverse edge ac while travelling through this intersection. Supposing c must be traversed in order to reach the destination, LHR will fail. The second possible failure occurs when a LHR traversal encounters the umbrella intersection first via node c in Figure 3 using the dashed line. LHR traverses the inside of the triangle $\triangle abd$ and exits without ever seeing edges that protrude from the outside of the triangle. As before, any such edges leading to the destination may be overlooked by an LHR traversal.

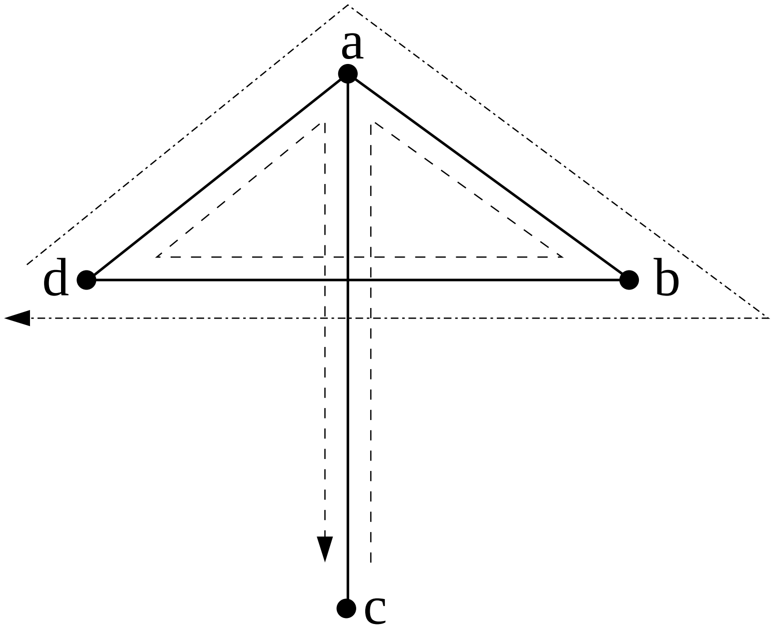


Figure 3: The 'bad' umbrella configuration prevents the success of LHR two ways.

The cause of both failures lies in the relationship between $\triangle abd$ and ac in Figure 3: There exists an edge from the triangle that is accessible only from inside the triangle. In other words, a traversal around the inside or the outside of the triangle fails to encounter all edges leading to the triangle. Both failures are solved by removing any of the edges that form the triangle. The easiest

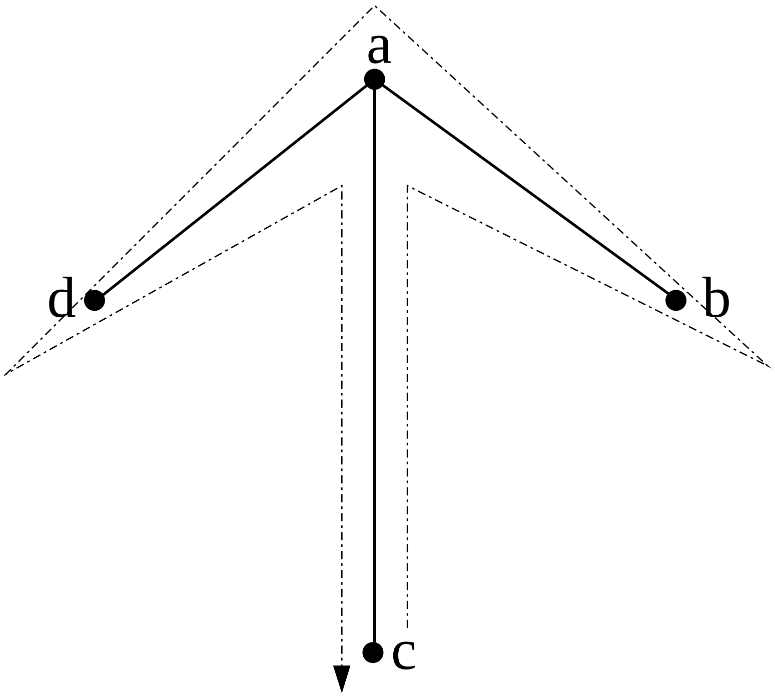


Figure 4: Removing prohibitive link *bd* allows LHR to traverse all edges.

of these to identify and remove from the network graph is the edge of the triangle that forms the intersection in the umbrella configuration. We call this the *prohibitive link*.

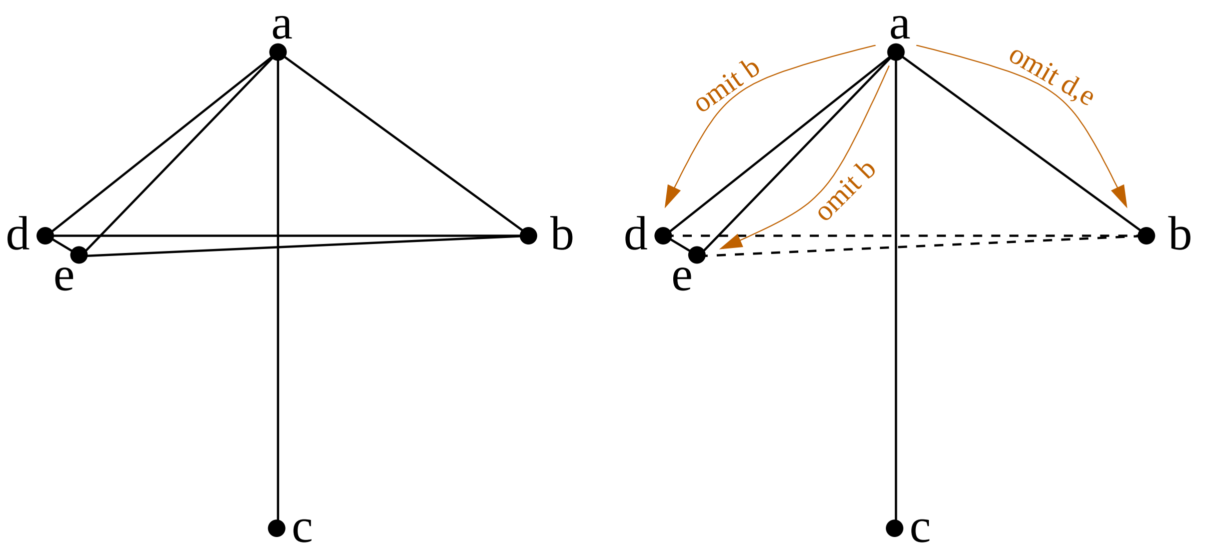
We revisit this subject and build a networking protocol in the next section. Before closing this section the outcome following a removal of the prohibitive link from the umbrella configuration is demonstrated in Figure 4. The configuration that was an intersection is reduced to a planar set of edges easily navigated by left- or right-hand rule.

4 Prohibitive-link Detection and Routing Protocol (PDRP)

We have enumerated all possible intersections in the unit-disc graph and identified the type of intersection with the link that prohibits successful delivery when routing according to right- or left-hand rule. In this section we present a prohibitive-link detection and routing protocol (PDRP) and show its correctness.

4.1 PDRP Overview

Most wireless protocols construct network graphs to guarantee delivery. The goal of PDRP is to remove from the network graph as few links as needed to guarantee routing. To better describe PDRP we assume a static graph where each node is assigned a coordinate in a 2-dimensional Euclidean system. We assume that the graph is connected and that all links are bi-directional. PDRP functions adequately in a mobile space provided that changes in position occur over a greater time-frame than is required to re-evaluate local prohibitive links and transmit local updates. In this work communication range is fixed and uniform across all nodes; a relaxation of this requirement is the subject of future work.



(a) Before detection phase.

(b) Following detection phase links db and eb omitted.

Figure 5: Local neighbourhood from viewpoint of node c, before and after the PDRP detection phase.

PDRP consists of two phases, detection and routing. The routing component operates similarly to the routing component in many established position-based routing protocols: PDRP routing consists of a greedy phase and a recovery phase. During normal operation nodes route in a greedy fashion and forward messages to the neighbour that most reduces the distance to the destination. Where no such neighbour exists a message is deemed 'stuck' in a local minima and is forwarded according to left- (or right-)hand rule. (The node initially selected is the first to appear left, or right, of the line segment from the current location to the destination.) The first node found that sits closer to the destination than the 'stuck' location returns to the greedy forwarding phase. The correctness of PDRP and its ability to guarantee delivery to the destination is discussed in Section 4.2.

During the PDRP detection phase each node inspects its neighbourhood. Using reported neighbour positions each node evaluates intersections within range and flags any three neighbours that compose an umbrella configuration, as described in Section 3.2. Once sufficient information is compiled a node sends a notification packet to the neighbours that anchor prohibitive links. Within this packet are a list of prohibitive links to be avoided during the recovery phase.

The detection phase is demonstrated in Figure 5. In Figure 5a node a determines that two intersections in its vicinity contain prohibitive links, those links being bd and eb. Nodes b, d, and e have no knowledge of node c's existence. The responsibility falls on node a to inform neighbours of their prohibitive links. Moving to Figure 5b node a instructs each of d and e to ignore their links to b during recovery; similarly node a instructs b to omit links to d and e.

By definition, the number of messages required to 'destruct' the network graph are fewer in number than current efforts such as the mutual witness protocol in GPSR. Alternatively notifications may be avoided entirely by either i) passing and evaluating over 2-hop neighbourhoods or ii) compactly embedding (in a fixed space) prohibitive links within recovery packets. We intend to solve this in the future by use of Bloom filters to compress prohibitive link information into recovery packets. In the next section we prove that PDRP functions correctly.

4.2 Statement of Correctness

Having identified removed prohibitive links in umbrella configurations, we show in this section that PDRP will successfully route a message between two nodes if a path exists. We remind our reader that during the routing phase of PDRP, any standard position-based routing technique consisting of greedy + face-routing recovery may be implemented.

The following argument progresses first by defining the graph embedding so that we might state our claim. We then show correctness by tracing a face-routing traversal within intersections of the embedding defined.

Definition 4.1 *Let G be an embedding of a graph. We define $G_{unu} = UDG(G) \cap NUmI(G)$, where $UDG(G)$ is the unit-disc graph over G and $NUmI(G)$ is the subgraph of G where umbrella intersections are removed.*

Proposition 4.2 *We claim that in G_{unu} a traversal, T, consisting of left-hand rule with memory, will find and traverse a unique face.*

Proof. We prove by induction on the neighbourhoods witnessed by T. Consider the first neighbourhood, k_0, visible to starting node v. If no intersection is visible to v then the next edge in T is trivial. If, however, an intersection exists in k_0 then it must be in the form depicted in either of Figures 6a or 6b (see Section 3.1 for proof).

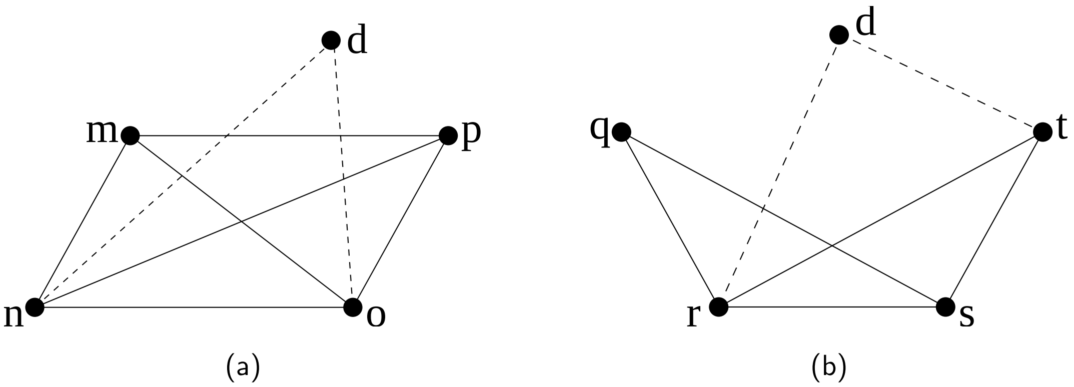


Figure 6: Once prohibitive links are omitted, two possible contentious configurations remain.

Case 1. Consider the intersection in Figure 6a. For any $v \in \{m, n, o, p\}$ and destination d, if $\overline{vd}$ intersects with no local edges (ie. $\overline{vd}$ does not pass through quadrilateral $(mnop)$) then the next left edge - and thus first edge in the current face - is trivial. If, however, $\overline{vd}$ does pass through $(mnop)$ as shown in Figure 6a then there are two cases:

$v = n$ The starting vertex is situated in the quadrilateral such that a single vertex sits left of $\overrightarrow{vd}$ and two vertices sit on the right. In Figure 6a this case is represented by $v = n$. Node n forwards to m. Both mo and mp intersect with nd, the line segment from the destination to the point where T started, so T will escape this neighbourhood when m chooses the next ccw edge from $\overrightarrow{mp}$.

$v = o$ The starting vertex is situated in the quadrilateral such that a single vertex sits right of $\overrightarrow{vd}$ and two vertices sit on the left. In Figure 6a this case is represented by $v = o$. Here, too, o forwards to m and m chooses the next ccw edge from $\overline{mp}$.

In either case, the face of interest begins at vertex m where a cycle, if traversed, will be declared.

Case 2. Consider the intersection in Figure 6b. Let starting node be $v \in q, r, s, t$ and destination d sit such that $\overline{rd}$ intersects $\overline{qs}$ and $\overline{sd}$ intersects $\overline{rt}$. The trivial case is $v = q$. Three cases remain:

$v = r$ The starting vertex is situated in the quadrilateral such that a single vertex sits left of $\overrightarrow{vd}$ and two vertices sit on the right. In Figure 6b this case is represented by $v = r$. r forwards to q where T will escape the neighbourhood. (Recall that when T intersects with $\overline{vd}$, T switches faces.) In this case the cycle will be detected at r.

$v = s$ The starting vertex is situated in the quadrilateral such that a single vertex sits right of $\overrightarrow{vd}$ and two vertices sit on the left. In Figure 6b this case is represented by $v = s$. s forwards along $\overline{sq}$ where T escapes the neighbourhood. On its return, T will detect a cycle at s since node t will avoid the edge $\overline{sq}$ since it was traversed previously.

$v = t$ The starting vertex is situated in the quadrilateral such that all three vertices sit left of $\overrightarrow{vd}$. In Figure 6b this case is represented by $v = t$. Here, T traverses $\overline{tr}, \overline{rq}$ before its escape from q. (Note that $\overline{qs}$ is not a valid edge since it intersects the edge previously traversed, $\overline{tr}$. T will detect this cycle at node t.

Assume now that for any neighbourhood, k_i, traversal T exits on the same face on which it enters. We show that for neighbourhood k_{i+1} traversal T exits on the same unique face on which it enters.

Referring once more to Figure 6, there are two types of neighbourhoods to consider. Those intersections whose endpoints join into a quadrilateral such as in Figure 6a require little consideration. For any entry point m, n, o, p on the quadrilateral, T will exit on the outside of this neighbourhood.

Similarly in Figure 6b, traversals entering on $\{q, r, s\}$ are trivial. We focus on traversals of T that reach node t. From t the next ccw edges in T are $\{\overline{tr}, \overline{rq}\}$ since $\overline{qs}$ intersects $\overline{tr}$. From q, T is forwarded along the next ccw edge. ▮

Corollary 4.3 *For any G_{unu}, a traversal, T, consisting of left-hand rule with memory, guarantees a path will be found provided a path exists, or complete the face if no path exists.*

*Proof.*We know that for a set of unique faces (ie. no intersections) in an embedding, that a left-hand traversal from source to destination is guaranteed to find a path provided one exists. Thus T, which finds sets of unique (non-intersecting) faces will find a path if it exists, or complete the face where no path exists. ▮

Finally, we note that traversal T requires no memory to succeed. A trace with no memory through all examples reveals that T will escape from any intersecting neighbourhood via the same egress links, albeit by traversing a few extra links.

PDRP overcomes drawbacks of existing methods. We show in the next section that an implementation of PDRP is also competitive with these methods.

5 Simulation Results

The previous section describes the PDRP protocol and asserts its correctness. We demonstrate the practical performance of PDRP via simulation in the sections that follow.

5.1 Experimental Design

We have implemented PDRP into the CLDP suite of position-based routing protocols [14] available for TinyOS [10, 26]. TinyOS is an event-driven operating system deployed on many commercial sensor networking products. Code written for TinyOS may be executed directly within TOSSIM [21], a simulator designed for debugging and evaluation of protocols before installation to sensor devices. The CLDP suite is a natural fit: In addition to CLDP it implements GPSR and face-routing with a variety of options; its implementation helps to ready PDRP for testing and improvement in real-world environments.

In our evaluation we compare PDRP versus CLDP and GPSR-MW. Recall from Section 2 that GPSR constructs a planar graph from available links. CLDP achieves planarity by destructing the network graph by probing links to identify and remove intersections. Likewise, PDRP destructs the network graph yet requires no probing and seeks to minimally disturb the graph.

GPSR simulations use the Gabriel Graph option for planarization. GPSR design and accomplishments serve as the foundation on which much more complex efforts have been built; it has long been considered the baseline for benchmark performance. In our simulations we use GPSR-MW which includes the mutual witness (MW) protocol for additional robustness. CLDP was developed much more recently to avoid the pitfalls that occur when the unit-disc graph assumptions are violated in real-world networks. PDRP, GPSR, and CLDP are similar in construction and operation yet differ in their design goals. A comparison in performance between these protocols is most appropriate.

Simulations are composed of 200 nodes placed uniformly at random within a square two-dimensional space. The communication range for all nodes is fixed at 180 units. Node density varies as we scale the network space by 100 units in the range from 1300 to 2000 units squared. Each simulation tests protocols over 50 runs using the same 50 topologies generated from non-overlapping random streams. This is sufficient to guarantee a 95% confidence interval. Our two primary performance measures are success rate and average path stretch. We additionally investigate the messaging overhead during setup. Each measure is defined and discussed within their relevant sections below.

5.2 Routing Setup and Success

The success rate is measured as the fraction of messages that are correctly transmitted between each pair of source-destination nodes. Figure 7 shows the success rate of each protocol as a function of network density described by average number of neighbours.

All protocols perform well though consistency appears to be highest among PDRP simulations. Note that our simulation results of CLDP differ slightly from results reported in [15]. In addition both CLDP and PDRP fall short of the expected 100% delivery rate that is predicted by their proofs of correctness. This is attributable to two phenomena. The first is collinearity. Left uncorrected, collinear links have the potential to confuse left- or right-hand rule in a manner that removes

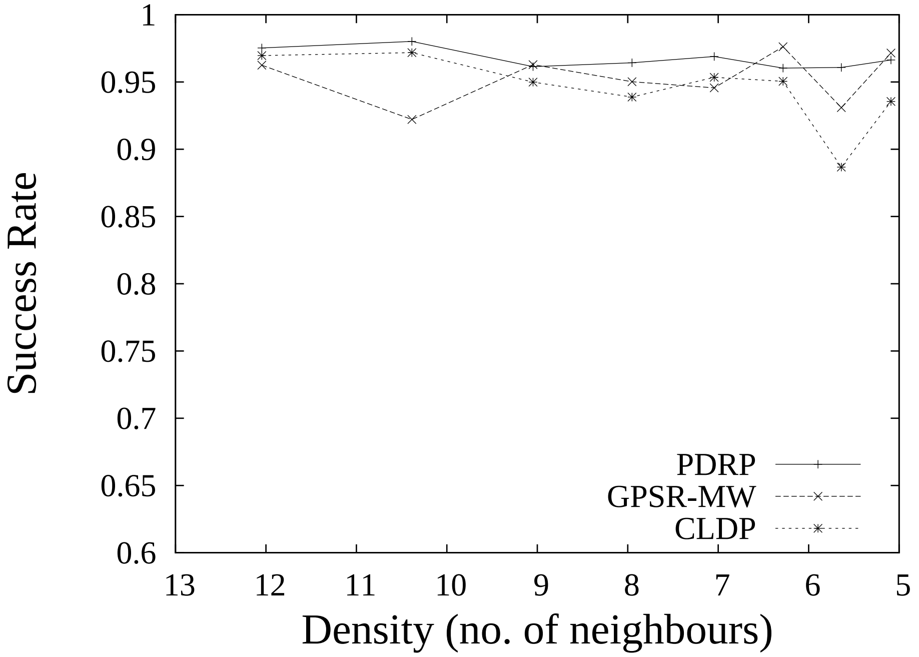


Figure 7: Routing Success.

routing guarantees. In Figure 7 collinearity is directly responsible for the dip that occurs in the GPSR-MW plot at $\tilde{1}0.5$ neighbours, as well as at $\tilde{6}$ neighbours in GPSR-MW and CLDP.

Referring again to Figure 7, PDRP seems less affected by collinear links. Further investigation reveals this to be an unexpected side-effect of the design goal underlying PDRP: to preserve the original network graph in a manner that allows delivery using position-based routing. In preserving as many links as possible PDRP maintains links that are omitted from the GPSR and CLDP network subgraphs. Hence there appears fewer collinear links since a greater number of links are seen by left- or right-hand rule in angular order in the PDRP subgraph.

The second reason we see a slightly less than 100% success rate is in the TinyOS implementation. There appears to be slight bugs about which we are in contact with the authors of the position-based routing suite for TinyOS.

An additional evaluation appears in Figure 8 which shows the number of messages that are exchanged during the setup phase of both PDRP and GPSR with the mutual witness protocol. (CLDP data has been omitted since the number of messages exchanged is many times higher than PDRP and GPSR.) In the current implementation both protocols search for relevant network

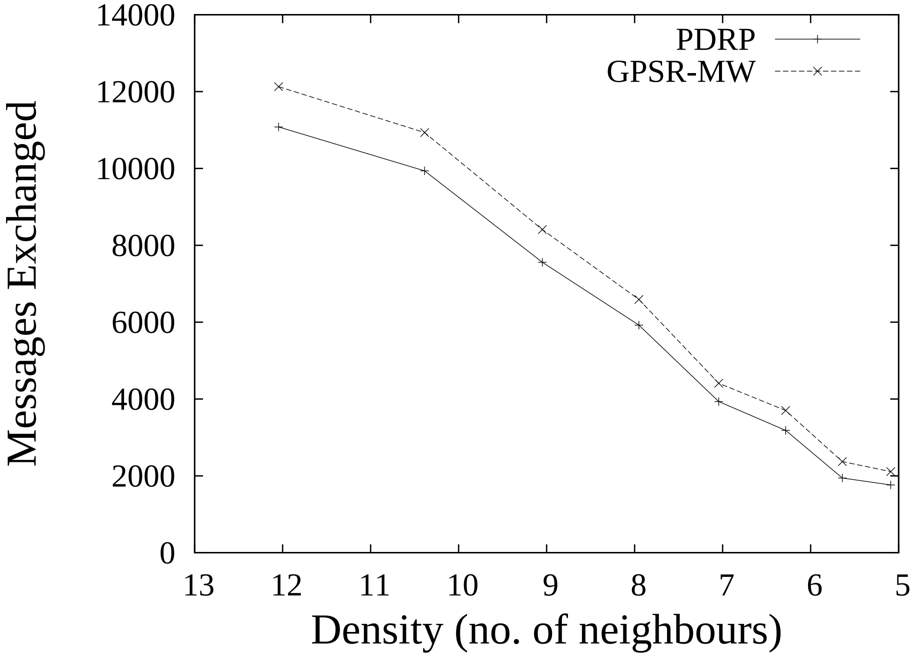


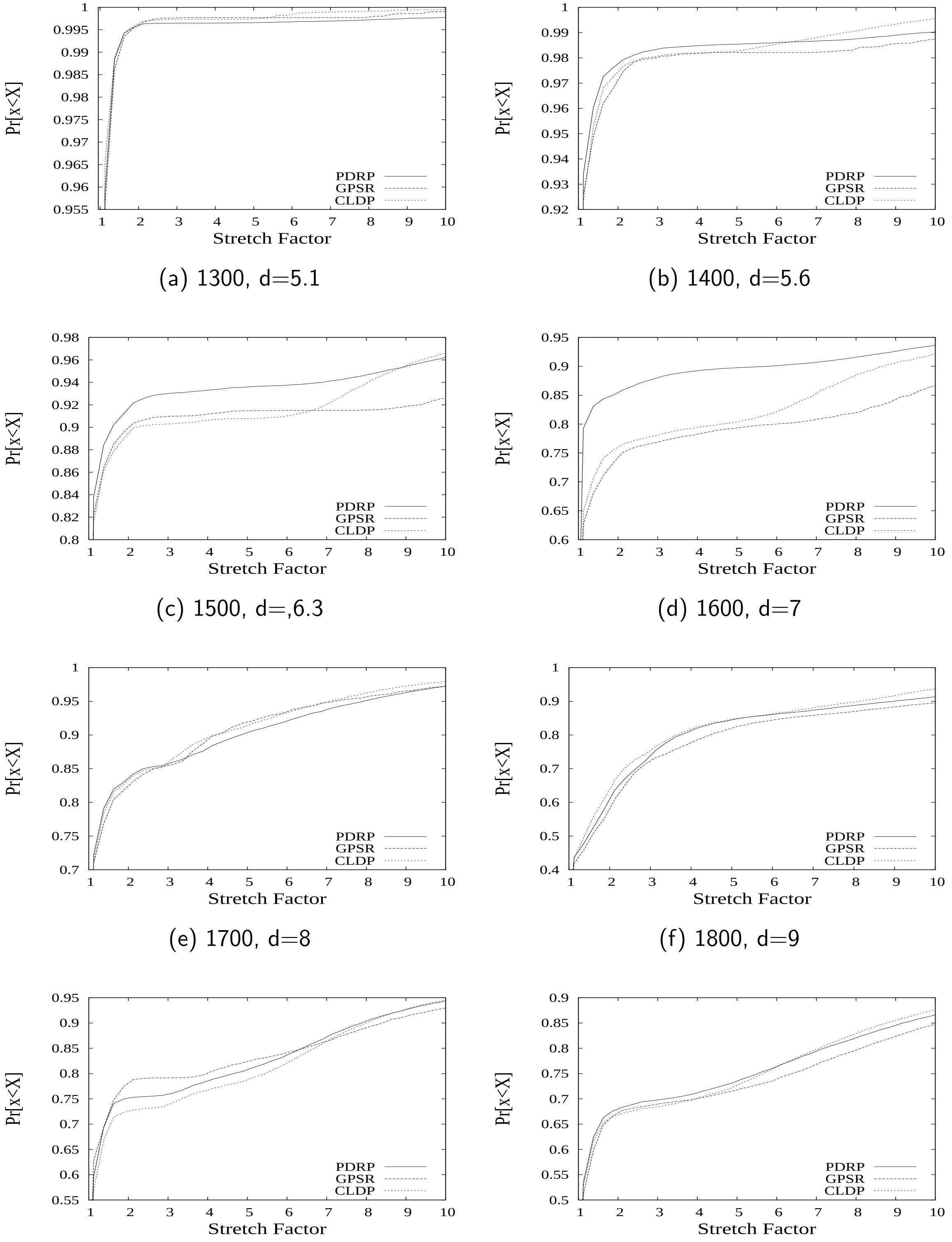
Figure 8: The number of messages exchanged during setup.

features upon receipt of each new 'HELLO' message. Updates are sent whenever there is a change in the local topology that affects routing. While there is room for optimisation, we can see that the messaging complexity of PDRP is on the order of GPSR-MW. In the future we intend to reduce this complexity further by compactly embedding prohibitive link information in the packet headers.

5.3 Routing Quality

We measure routing quality using path stretch factor. The stretch factor describes the ratio between the number of hops in the shortest path and the number of hops traversed using the routing scheme in question. The cumulative distribution functions generated by stretch factors appear in Figure 9. Subplots show the cdf as revealed over networks of increasing density. We emphasize to our reader the differences in scale (over the y-axes) between plots so that appropriate levels of detail may be viewed for each set of simulations.

The difference in performance between GPSR, CLDP, and PDRP in the sparsest networks, shown in Figures 9a and 9b, is near indistinguishable. The reduced likelihood of intersecting links that occurs with sparse networks means that all three protocols are more likely to choose the same



(a) 1300, d=5.1

(b) 1400, d=5.6

(c) 1500, d=,6.3

(d) 1600, d=7

(e) 1700, d=8

(f) 1800, d=9

(g) 1900, d=10.4

(h) 2000, d=12

Figure 9: Edge proximity distributions reveal the proximity to the network edge of nodes that declare boundary status.

path. Furthermore node sparsity reduces the number of potentially good routing choices so each protocol is more likely to route along the shortest path. Referring now to Figures 9e through 9h we can make two observations. The first observation is that routing quality amongst all three protocols continues to to behave in a manner that is (statistically) indistinguishable. Second, we can see a clear degradation in path quality that is directly attributed to the increase in frequency of intersections and their potential to separate the routing path from the shortest path during recovery modes.

There does appear, however, a gap of particular interest that separates PDRP from GPSR and CLDP. It is visible in Figure 9d which depicts the cdf of path stretch factor where the number of neighbours per node is approximately 7. Further investigation reveals that the improved routing quality provided by PDRP stems from its propensity to leave intersecting links intact during the recovery phase: PDRP often routes along direct links between neighbours whereas the GPSR-family of protocols will route along multiple links between the same neighbours. Why this separation is much more clear when neighbourhoods are approximately 7 nodes in number rather than the more dense or sparse nodes remains an open question and requires further investigation.

6 Conclusions

In this paper we have explored an new approach to graph construction for successful forwarding in position-based routing. It is instructive to compare this approach with previous work.

Like the face-routing family of protocols, the success of PDRP relies upon a recovery phase that implements a form of left-hand rule. Traditionally, the success of face-routing schemes relies on the assumption that the underlying graph is planar. The planar graph was chosen because, using left-hand rule, a path to the destination will always be found (if a path exists). This is restrictive; local constructions of planar graphs risk inaccuracies, while co-operative (or global) constructions are resource intensive. In either case there has yet to appear an examination of the challenges that face left-hand rule in the presence of intersections.

By contrast, the approach taken in this work was to enumerate the configurations that form an intersection in the network graph. We then scrutinised each with a left-hand rule traversal so as to isolate the 'bad' configurations from which left-hand rule is unable to recover. In doing so we recognised the existence of a prohibitive link that has the potential to conceal other viable links from a left-hand rule traversal. We then presented PDRP, a protocol that detects and avoids the prohibitive link to successfully deliver packets. It operates locally and, unlike planarization methods, omits only nonessential links.

Our simulation results demonstrate that PDRP performance is similar to, and in some cases out-performs, current face-routing schemes. Compared against CLDP and GPSR with the mutual witness protocol, the success rate of PDRP appeared more consistent across networks of varying density. The availability of additional edges to left-hand rule during the recovery phase allowed PDRP to avoid the trends common to both GPSR and CLDP. Moreover, in all cases the stretch factor of paths generated by PDRP was competitive with both GPSR and CLDP. In networks where neighbouring nodes averaged 7, PDRP shows noticeable improvement over GPSR and CLDP; the underlying cause requires further analysis.

We have implemented PDRP into the geographic routing suite for TinyOS and are pleased to make it available upon request. In the future we hope to remove the unit-disc assumption. Then, using the approach presented in this paper we expect to augment PDRP for general case networks where communication error and non-uniform range is commonplace.

References

[1] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in *ad hoc* wireless networks. In *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM)*, 1999.

[2] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in *ad hoc* wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[3] D. Chen and P. Varshney. A survey of void handling techniques for geographic routing in wireless networks. *Communications Surveys & Tutorials, IEEE*, 9(1):50–67, First Quarter 2007.

[4] D. Chen and P. K. Varshney. On-demand geographic forwarding for data delivery in wireless sensor networks. *Elsevier Computer Communications*, 30(14-15):2954–2967, 2007.

[5] H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *the 12th annual international conference on Mobile computing and networking (Mobicom)*, pages 390–401, 2006.

[6] H. Fler, J. Widmer, M. Ksemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier's Ad Hoc Networks*, 1(4):351–369, November 2003.

[7] Y.-J. K. R. Govindan, B. Karp, and S. Shenker. Lazy cross-link removal for geographic routing. In *the 4th international conference on Embedded networked sensor systems (SenSys)*, pages 112–124, 2006.

[8] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli. Evaluating the limitations of and alternatives in beaconing. *Elsevier Ad Hoc Networks*, 5(5):558–578, 2007.

[9] M. Heissenbttel, T. Braun, T. Bernoulli, and M. W. A. Blr: Beacon-less routing algorithm for mobile ad-hoc networks. *Elseviers Computer Communications Journal (Special Issue*, 27:1076–1086, 2004.

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *SIGPLAN Notices*, 35(11):93–104, 2000.

[11] H. Kalosha, A. Nayak, S. Ruhrup, and I. Stojmenovic. Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks. In *IEEE 27th Conference on Computer Communications (INFOCOM).*, pages 346–350, Pheonix, AZ, USA, April 2008.

[12] B. Karp. Challenges in geographic routing: Sparse networks, obstacles, and tra ffic provisioning. presented at DIMACS Workshop on Pervasive Networking, May 2001.

[13] B. Karp and H. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM MobiCom*, Boston, MA, September 2000.

[14] Y.-J. Kim. CLDP+GFR ver-0.2 for the motes. http://enl.usc.edu/software.html, 2005.

[15] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, USA, May 2005.

[16] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC*, pages 34–43, 2005.

[17] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Ad-Hoc Routing: Of Theory and Practice. In *22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, Boston, Massachusetts, USA, July 2003.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically Optimal Geometric Mobile Ad-Hoc Routing. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, Atlanta, Georgia, September 2002.

[19] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Annapolis, Maryland, USA, June 2003.

[20] B. Leong, B. Liskov, and R. Morris. Geographic routing without planarization. In *Proceedings of the 3rd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, USA, May 2006.

[21] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 126–137, 2003.

[22] C. Lochert, M. Mauve, H. Fler, and H. H. rtenstein. Geographic Routing in City Scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 9(1):69–72, January 2005.

[23] J. Sanchez, R. Marin-Perez, and P. Ruiz. Boss: Beacon-less on demand strategy for geographic routing in wireless sensor networks. In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS).*, pages 1–10, October 2007.

[24] C. Schindelhauer, K. Volbert, and M. Ziegler. Geometric spanners with applications in wireless networks. *Computational Geometry: Theory and Applications*, 36(3):197–214, 2007.

[25] K. Seada, A. Helmy, and R. Govindan. On the effect of localization errors on geographic face routing in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks (IPSN)*, pages 71–80, 2004.

[26] Tinyos source, software, and documentation. http://www.tinyos.net.

[27] G. Zhao, X. Liu, M.-T. Sun, and X. Ma. Energy-efficient geographic routing with virtual anchors based on projection distance. *Elsevier Computer Communications*, 31(10), 2008.