Practical and Robust Geographic Routing In Wireless Networks

Young-Jin Kim Ramesh Govindan Brad Karp Scott Shenker

August 28, 2004

Abstract

Existing geographic face routing algorithms use planarization techniques that rely on the unit-graph assumption, and thus can exhibit persistent routing failure when used with real radios, whose connectivity violates that assumption. In this paper, we describe the Cross-Link Detection Protocol (CLDP), which enables provably correct geographic routing on *arbitrary* graphs. Our simulations show that the protocol is practical: it incurs low overhead and exhibits low path stretch in wireless networks.

1 Introduction

Geographic routing protocols for wireless ad hoc networks are highly attractive because they have been shown to scale better than other alternatives. They require that nodes store only a list of their immediate single-hop neighbors, and thus require per-node state independent of the total number of nodes in the network, and dependent only on the network's density. Practical instantiations of such protocols have been shown to achieve high packet delivery success rates even under highly dynamic network topologies such as occur on mobile, wireless networks. Moreover, they do so while incurring relatively little routing protocol traffic overhead [11]. More recently, geographic routing algorithms have been proposed for use as a routing primitive for static sensor networks, and as building blocks for data storage and flexible query processing in sensor networks [20, 17].

There is a very broad literature on geographic routing algorithms, particularly on the sub-class that uses *face routing* on a planar subgraph [7]. This body of literature is built upon graph planarization algorithms that are amenable to distributed implementation. More specifically, these algorithms rely purely on neighbor location information to determine whether links to neighbors belong on the planarized subgraph or not. A packet is delivered from source to destination by successively traversing the faces on the planar subgraph that intersect the line between them. Early work by Bose *et al.* [2] and Karp and Kung [11] described the planarization algorithms, as well as practical instantiations of these algorithms for ad hoc wireless networks (*e.g.*, GPSR). An extensive body of subsequent work (including GOAFR+ [14] and its many variants) has focused on proving various properties of these algorithms, and improving the worst-case behavior of face routing. A common assumption made by this body of literature is that connectivity between nodes can be described by unit graphs.[1] In such graphs, a node is always connected to all nodes within its fixed, "nominal" radio range, and never connected to nodes outside this range.

Real radios violate the unit graph assumption more often than not. As many recent studies show [1, 5, 25, 27], connectivity depends not only on the distance from the transmitter but also on the environment (the presence of radio-opaque obstacles, multi-pathing, *etc.*). These empirical observations suggest that the unit graph (or even the quasi unit-disk) is not a reasonable model for radio connectivity regardless of the kind of radio (802.11 or the low-power radios in use in sensor networks today). Furthermore, in sensor network applications, inaccurate location information might result in violations of the unit graph assumption: *e.g.*, a neighbor might report its location as being within the nominal radio range of another neighbor, even when it is not.

We have found that these violations can result in pathologies in the planarization process. Specifically, three kinds of pathologies result when the unit graph assumption is violated: a link in the planar subgraph is removed when it should not have been (disconnected links); the nodes at the two ends of a link disagree on whether the link belongs to the planar graph or not (unidirectional links); or, two crossing links exist in the planar subgraph (crossed links). These pathologies, in turn, can result in *persistent* routing failures in the network, where geographic routing fails to find a path for at least one source-destination pair. We also show that a previously proposed "fix" to these planarization techniques, the mutual-witness procedure [9, 10, 22], does not eliminate all instances of routing failure.[2]

In this paper, we discuss the design of a distributed

[1] Some recent literature [15] has relaxed this assumption to allow for "quasi" unit disk graphs, which always exhibit connectivity within a short radius; exhibit probabilistic connectivity within an enclosing ring-shaped region; and exhibit no connectivity beyond this ring. We discuss these graphs briefly in Section 2.

[2] As we show in our simulations, the mutual witness technique does very well, achieving upwards of 99% packet delivery success even in networks with an unrealistically large number of obstacles. However, we believe that it is unreasonable for a routing protocol to permanently partition even a few source-destination pairs.

Cross-link Detection Protocol (CLDP) that, *given an arbitrary connected graph*, produces a subgraph on which face traversal cannot cause a routing failure. In CLDP, each node probes the faces on which each of its links sits to determine if there exists a crossing link. Crossing links are eliminated only when doing so would not disconnect the resulting subgraph. Packets are then routed using greedy and perimeter-mode routing as in GPSR. Other face routing techniques [14] can be used as well; such techniques would not impact the correctness of CLDP, but may affect its performance.

We prove under some idealized, but still general, assumptions that CLDP cannot cause a routing failure in an arbitrary connected graph. We then use detailed packet-level simulations to validate the performance of CLDP both on wireless networks with many obstacles and on random graphs. Our simulations find *no* routing failures in *all* the cases we study, and show that CLDP has reasonable path stretch and low overhead and convergence times. We conclude that CLDP's provable correctness and measured efficiency bring geographic routing protocols within the realm of practicability for real radio networks.

2 Preliminaries and Related Work

We now review prior work in geographic routing protocols, and describe the essentials of the workings of geographic routing that provide the context for our work.

There is a tremendously broad literature on geographic routing: from initial sketches suggesting routing using position information [12, 3]; to the first practical, detailed proposals, including GFG [2], GPSR [11], and the GOAFR+ family of algorithms [14]; to refinements of these proposals for efficiency [6], robustness under real network conditions [22, 15], and even routing geographically when node location information is unavailable [19, 18].

We now describe the shared characteristics of the GFG, GPSR, and GOAFR+ algorithms, and hereafter refer to this family of algorithms simply as *geographic routing.*[3]

Geographic routing schemes use *greedy routing* where possible. In greedy routing, packets are stamped with the *positions* of their destinations; all nodes know their own positions; and a node forwards a packet to its neighbor that is geographically closest to the destination, *so long as that neighbor is closer to the destination. Local maxima* may exist where no neighbor is closer to the destination. In such cases, greedy forwarding fails, and another strategy must be used to continue making progress toward the destination. In particular, the packet must only find its way to a node closer to the destination than the local maximum; at that point, greedy routing may once again make progress.

In the case where a network graph has no crossing edges[4]—that is, the graph is *planar*—the geographic routing schemes recover similarly, by *face routing*. Note that a planar graph consists of *faces*, enclosed polygonal regions bounded by edges. Geographic routing schemes use two primitives to traverse planar graphs: the *right-hand rule*, and *face changes*. The right-hand rule tours a face endlessly in a cycle, and can thus be used to walk a face. Figure 1 shows an example of the rule, which dictates that upon receiving a packet on a link, the receiving node forwards that packet on the first link it finds after sweeping counter-clockwise about itself from the ingress link.

Consider the planar graph in Figure 2, in which the source node S and destination node D are indicated. Observe that the line segment $\overline{SD}$ *must* cut a series of faces in the planar graph; these faces are numbered and bordered in bold. Geographic routing algorithms exploit this property by successively walking the faces cut by this line. That is, they use the right-hand rule to tour a face. While walking a face, upon encountering an edge that crosses the line segment $\overline{SD}$ at a point closer to D than the point at which the current face was entered, geographic routing algorithms perform a *face change:* they begin walking the bordering face that is next along the line segment $\overline{SD}$.[5] The numbering of faces in Figure 2 shows the order in which faces are traversed from S to D on that planar graph. Should a face be toured in its entirety without discovering an edge that crosses line segment $\overline{SD}$ at a point closer to D than the point at which the current face was entered, face routing fails. On a planar graph, such a loop on a face only occurs when the destination is disconnected.

Note that if the graph is not planar, face routing may fail. Figure 3 shows an example graph on which this pathology occurs. In this example, D is located physically in the interior of a face, but is only connected to the rest of the network graph by an edge that crosses this enclosing face. Face routing walks successive faces cut by the line from S to D, until it reaches the face enclosing D, whose first edge crosses line segment $\overline{SD}$ at point p. The right-hand rule then tours this face in its entirety, but fails to find an edge that crosses line segment $\overline{SD}$ at a point closer to D than p. Thus, face routing fails.

Wireless networks' connectivity graphs typically contain many crossing edges. A method for obtaining a planar subgraph of a wireless network graph is thus needed; greedy routing operates on the full network graph, but to work correctly, face routing must operate on a planar subgraph of the full network graph. What is required is a *pla-*

[3] We note that there exist other routing algorithms that make use of position information, such as LAR [13], but we restrict our view to algorithms in which a node forwards to a single neighbor on the basis of geographic information.

[4] We refer to links and edges interchangeably throughout the paper.

[5] Other face-change rules are possible, including changing faces at the edge whose crossing of $\overline{SD}$ is the *closest* such crossing to D on the current face. We use the first crossing, not best crossing, throughout this paper; this choice is known to be average-case efficient, and has been refined [14] to be worst-case optimal. We return to this point in Section 4.1.

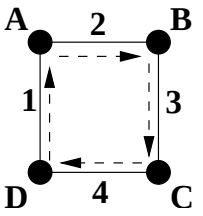

Figure 1: Right-hand rule. *A* sweeps counterclockwise from link 1 to find link 2, forwards to *B*, *&c.*

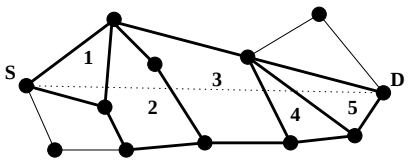

Figure 2: The faces progressively closer from *S* to *D* along line segment $\overline{SD}$, numbered in the order visited. Faces cut by $\overline{SD}$ are bordered in bold.

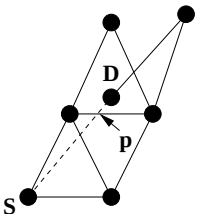

Figure 3: Example of face routing failure on non-planar graphs. There is no point closer to *D* than *p* on the face enclosing *D*.

narization technique that is simply implementable with an asynchronous distributed algorithm.

Geographic routing algorithms planarize graphs using two planar graph constructs that meet that requirement: the Relative Neighborhood Graph (RNG) [24] and the Gabriel Graph (GG) [4]. The RNG and GG give rules for how to connect vertices placed in a plane with edges based purely on the positions of each vertex's single-hop neighbors. Both the RNG and GG provably yield a connected, planar graph so long as the connectivity between nodes obeys the *unit graph assumption:* for any two vertices *A* and *B*, those two vertices *must* be connected by an edge if they are less or equal to some threshold distance *d* apart, but *must not* be connected by an edge if they are greater than *d* apart. We shall refer to *d* as the *nominal radio range* in a wireless network; the notion is that all nodes have perfectly circular radio ranges of radius *d*, centered at their own positions.

The unit graph assumption is quite intuitive for wireless networks. The simplest ideal radio model is one where all transmitters radiate fixed transmission power perfectly omnidirectionally; receivers can discern all transmissions properly when they are received with above some threshold signal-to-noise ratio; and radio transmissions propagate in free space, such that their energy dissipates as the square of distance. Under that idealized model, there indeed exists a nominal radio range.

We briefly state the definitions of the GG and RNG, as we shall refer to them repeatedly in Section 3. The planarization process runs on a *full graph*, which includes *all* links in the radio network, and produces a *planar subgraph* of the full graph. We assume that each node in the network knows its single-hop neighbors' positions; such neighbor information is trivially obtained if each node periodically transmits broadcast packets containing its own position. Consider an edge in the full graph between two nodes *A* and *B*. Both *A* and *B* must decide whether to keep the edge between them in the planar graph, or eliminate it in the planar graph. Without loss of generality, consider node *A*. Both for the GG and RNG, node *A* searches its single-hop neighbor list for any *witness* node *W* that lies within a particular geometric region. If one or more witnesses are found, the edge (A,B) is eliminated in the planar graph. If no witnesses are found, the edge (A,B) is kept in the planar graph. For the GG, the region where a witness must exist to eliminate the edge is the circle whose diameter is line segment $\overline{AB}$. For the RNG, this region is the *lune* defined by the intersection of the two circles centered at *A* and *B*, each with radius $|\overline{AB}|$. We show these two regions in Figure 4.

Under the unit graph assumption, it is known that for a clustering of points in the plane, the set of edges in the Euclidean minimum spanning tree over those points is a subset of the set of edges in the RNG [24]. The edges in the RNG are in turn a subset of those in the GG; the intuition for this relationship lies in the relative sizes of the lune and circle regions. Finally, the set of edges in the GG is a subset of that in the Delaunay triangulation over the set of points [23]. These relationships dictate that the GG and RNG are both connected (so eliminating crossing edges cannot disconnect the network!) and planar, as desired.

Note that if the network graph *violates* the unit graph assumption, the RNG and GG can produce a *partitioned* planarized graph [9], one that contains asymmetric (unidirectional) links, and even one that is not planar. An example of a partitioning for the RNG appears in Figure 5. Here, there is no link between *A* and *V*, and none between *B* and *W*, though these links are shorter than the nominal radio range. Nodes *A* and *B* see witnesses *W* and *V*, respectively, though neither witness provides transitive connectivity. Both *A* and *B* conclude they should remove edge (A,B) in the planarized graph, and a partition results. Similar cases are possible in the GG.

We observe that whether radio graphs conform to the unit-graph assumption is a question of great importance, as partitioning the planarized graph used in face routing will cause routing failures. In the next section, we explore in detail the many reasons real radio networks violate the unit graph assumption, and give detailed examples of the pathologies these violations create in the GG and RNG.

Recently, Kuhn *et al.* have investigated relaxing the unit-graph assumption to improve the robustness of the GG planarization [15]. In the *Quasi-Unit Disk Graph*

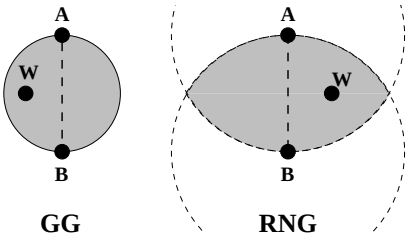

Figure 4: Geometric definitions of the GG and RNG. A witness must fall within the shaded circle (GG) or the shaded lune (RNG) for edge (A,B) to be eliminated in the planar graph.

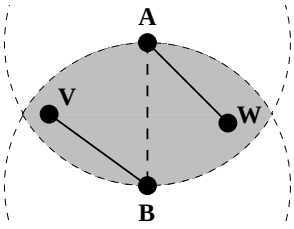

Figure 5: The RNG partitions a non-unit graph; edge (A,B) is eliminated.

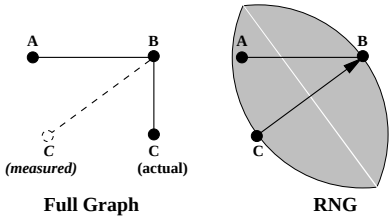

Figure 6: Localization errors can cause the RNG to produce a planar graph containing an asymmetric link.

they propose, the nominal radio range is normalized to 1. Links *may not* exist between nodes greater than distance 1 apart, and links *must* exist between nodes less than a parameter d apart. For nodes between d and 1 distance apart, links may or may not exist; it's in this region where Quasi-Unit Disk Graphs are a more general class than unit graphs. Kuhn *et al.* provide an algorithm for replacing "missing" links between d and 1 in length with *virtual* links, that are essentially tunnels through multiple existing links. They show that the GG planarization succeeds on this augmented graph without partitioning it. Their analysis shows that this technique only is scalable when $d \geq 1/\sqrt{2}$; for lesser values of d (for which the unit-graph assumption is progressively relaxed further) virtual links may be comprised of increasingly long paths of physical hops.

We close by noting that there is a much wider swathe of theoretical literature concerning geographic routing; Stojmenovic [7] offers a comprehensive survey of the many contributions in that domain.

3 Face Routing Using Real Radios

In this section, we show that *the RNG and GG, given realistic, connected radio graphs, do not always produce graphs on which face routing succeeds.*

3.1 Non-uniformity in Radio Ranges

Central to the notion of radio range in real deployments is the definition of a "link." In wired networks, there is no ambiguity: a link exists if a physical cable runs between two nodes. In wireless networks in which nodes use omnidirectional antennas, however, a link is more fuzzily defined: each receiver experiences a different bit error rate, and thus a different packet loss rate, when a transmitter sends a packet. Thus, the loss rate between a pair of nodes determines whether a link exists between them. We adopt the view that above a threshold loss rate, links are viewed as non-existent; links with lower loss rates than this threshold exist.[6] Under this practical link existence model, the crux of the validity of the unit-graph assumption is whether the loss rate between a pair of nodes is perfectly determined by the distance between them.

Measurement studies of deployed wireless networks provide mounting evidence of marked non-uniformity in radio ranges. Biswas and Morris [1] measure loss rates between pairs of hosts on an 802.11 *rooftop network* deployed in buildings spread throughout an urban setting. Their measurements reveal that links with low loss rates are often far longer than the expected nominal radio range, and conversely, that geographically proximal nodes often are connected by links with extremely high loss rates (*i.e.*, that these links effectively do not exist). Other studies also provide qualitatively similar evidence for sensor network radios [5, 25] in a variety of environments [27]. All of these studies also confirm the significant presence of link asymmetries. In sum, these measurements indicate that in indoor, urban outdoor, and habitat outdoor environments, for any reasonable "nominal radio range" one might choose, there both exist links longer than that threshold length, and are missing links shorter than that threshold length.

Many radio phenomena can contribute to the prevalence of non-uniform radio ranges. A *radio-opaque obstacle* may attenuate transmissions so severely that it breaks a link, even when the two nodes are closer to each other than the unit graph threshold distance. *Multipath interference* resulting from reflection of radio waves by objects in the environment can "delete" links within the nominal radio range when the original and reflected waves combine destructively. *Asymmetric links*, generally assumed to be caused by differences in transceiver calibration, also violate the unit-graph assumption which assumes bi-directional communication. Finally, *non-*

[6] Recent work investigates relaxing this threshold view of link existence in wireless networks [1], in favor of considering links with a very wide range of loss rates when routing, and choosing routes on the basis of link loss rates. This approach has been shown to increase throughputs between source-destination pairs *vs.* the loss-threshold approach, but is beyond the scope of our present investigation.

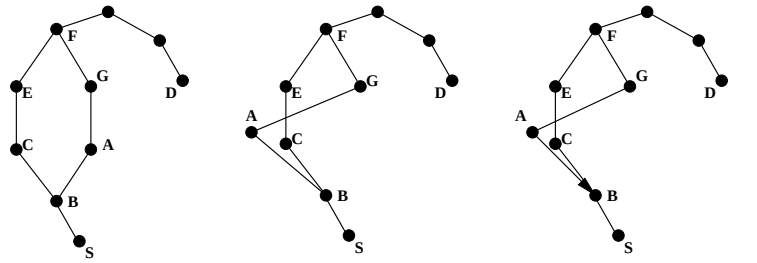

Figure 7: Localization errors can cause the RNG to produce a non-planar graph (after Seada *et al.* [22]).

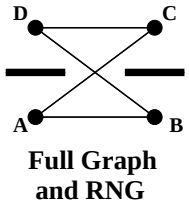

Figure 8: Mutual Witness leaves crossing edges in the graph produced by the RNG (and GG).

circular antenna emissions, often intrinsic to the physical construction or the deployment orientation of antennas, create the possibility of non-unit network graphs.

Localization errors can also violate the unit-graph assumption. Under mobility, a node's position estimate for itself may be out-of-date. Even when a node is stationary, errors in positions produced by GPS or by ad-hoc localization systems [21] are the rule. Two nodes may have position errors that place them within nominal radio range of one another, but in reality, they are not (even assuming perfectly circular radio ranges).

At least one of these causes for non-uniform radio ranges is easily detected, Asymmetric links can be easily *blacklisted:* if each node announces its neighbor list in a broadcast packet, receivers can eliminate neighbors in whose neighbor lists they do not appear [26]. In the remainder of this paper, we consider only symmetric radio links in the full graph.

3.2 RNG and GG on Non-Unit Graphs

Let us now examine the behavior of the RNG and GG on non-unit full graphs. We separately consider two classes of pathologies: irregular radio range pathologies, which essentially "delete" links expected to exist given the unit-graph assumption; and localization error pathologies. In our discussion, we use only the RNG in our examples, but all pathologies discussed apply to both the RNG and GG, because they use the same underlying witness mechanism in planarization.

Irregular Radio Ranges These pathologies subsume obstacles, multi-path interference, and non-circular antenna emissions, all of which can cause omission of unit-graph links in the full graph. In Section 2 (Figure 5), we gave an example in which the omission of two links in the full graph that are shorter than the nominal radio range causes the RNG to produce a planar but partitioned graph. We observe further that if only *one* of the two witnesses were present (*e.g.*, assume node W is not present in the graph), the resulting planar graph would contain a unidirectional link between A and B. Both partitions and unidirectional links in the planarized graph can cause routing failures for face routing.

Localization Errors Localization errors can cause the RNG to produce a planar graph with asymmetric links from the full graph. An example of this pathology appears in Figure 6. The full graph topology is given on the left; solid lines indicate links connected in the full graph. Here, because of localization error, node C believes it is located below node A, when in actuality it is located below node B. The RNG appears on the right: B will eliminate edge (B,C) in planarization, because it believes A is a witness; C keeps edge (C,B) as it has no link to A. Note that this case is isomorphic to one in which node C is truly located beneath node A, but an obstacle exists between nodes A and C.

Moreover, localization errors can cause the RNG to produce a *non-planar graph*; we present such a case in Figure 7. Here, all nodes know their correct positions but node A. The leftmost topology shows the *true* positions of all nodes in the full graph, which is already planar. The center topology shows the full graph with A's errored position. The right topology shows the resulting RNG. There exist an asymmetric link from A to B in the RNG, and a crossing edge in the RNG, *despite the planarity of the nodes' true positions in the full graph.* That is, the right-hand rule chooses a next hop using the erroneous position information of A. Face routing from S to D on the RNG will take the tour $S \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow A \rightarrow B \rightarrow S$, at which point a loop has occurred, and face routing fails.

3.3 A Potential Fix: Mutual Witness

Motivated by the fragility of the RNG and GG planarizations on real wireless network graphs, increasing attention has been paid to improving their robustness. One technique, *mutual witness*, has received attention in the literature recently [9, 10, 22]. When node A considers whether to keep edge (A,B) from the full graph in the RNG or GG planar graph, mutual witness dictates that A only eliminate edge (A,B) if there exists at least one witness in the

Planarization Type	Phenomenon	Partition	Crossing Edges	Asymmetric Edges
RNG / GG	Non-uniform Radio Range	√	√	√
	Localization Error	√	√	√
MW and RNG / GG	Non-uniform Radio Range	√	√	
	Localization Error	√	√	

Table 1: Pathologies in graphs produced by RNG and GG, and mutual-witness variants, as a function of features in the full graph.

RNG or GG region that is visible *both* to A and B. This fact may be directly verified with local communication: if all nodes broadcast their neighbor lists (only a single hop), then all nodes may verify whether a particular neighbor shares a particular other neighbor.

The intuition for this mutual witness rule is that it preserves connectivity: edges are only eliminated in the planar graph if a transitive path through a witness is explicitly verified, rather than relying on the location of the witness (the unit-graph assumption) to assure such a transitive path's existence. Indeed, mutual witness does preserve connectivity. Consider the example from Figure 5. Because neither V nor W is a witness that is shared both by A and B, edge (A,B) will be preserved, and the planar graph will remain connected.

Unfortunately, mutual witness suffers from another ill; on some non-unit graphs, it will *leave crossing edges* in the graph produced by the RNG and GG. Figure 8 shows just such an example. Obstacles block links (A,D) and (B,C). Two crossing edges remain because there are no witnesses in common between A and C or between B and D. Yet either of the two crossing edges could be removed, and the output of the RNG would then be properly planar. Thus, mutual witness cannot render all non-unit graphs planar.

We summarize our findings from this section in Table 1. We have found example topologies that elicit all these pathologies in the RNG and GG, and in their mutual-witness variants, but omit them in the interest of brevity. In sum, we know of no efficient, distributed planarization method that produces graphs on which geographic routing must succeed from *real* radio network graphs. In the next section, we present a planarization method that *provably* produces graphs on which geographic routing always succeeds, for *arbitrary* undirected connected graphs.

4 Cross-Link Detection Protocol

Having established that face routing can fail with existing planarization techniques when the unit-graph assumption is violated, we now proceed to describe the Cross-Link Detection Protocol (CLDP). Our exposition of CLDP's various mechanisms is informal and uses simple topologies as examples or counter-examples. The next section presents a more formal analysis of CLDP's correctness.

4.1 CLDP Overview

To describe the essential ideas behind CLDP, we start by considering a static graph consisting of several nodes and links. We make no assumptions about the connectivity of this graph (*i.e.,* to which other nodes a given node may be connected). However, we do assume that nodes in the graph are assigned positions in some 2-dimensional coordinate system, that the graph is connected, and that all the links are bi-directional. We also make several other idealized assumptions (like link-serialized execution of the protocol) in the rest of this subsection in order to simplify exposition. We will return a bit later to consider the applicability of CLDP to wireless networks: in particular, we will consider the impact of node and link dynamics, and present a truly distributed realization of CLDP.

The high-level idea behind CLDP is simple: each node, in an entirely distributed fashion, *probes* each of its links to see if it is *crossed* (in a geographic sense) by one or more other links. A probe initially contains the locations of the endpoints of the link being probed, and traverses the graph using the right-hand rule. For example, in Figure 9, consider a probe originated by node D for the link (D,A). It contains the geographic coordinates for D and A, and traverses the graph using the right-hand rule, as shown by the dashed arrows. When the probe is about to traverse the edge (B,C), node B "notices" that this traversal would cross (D,A); B records this fact in the probe so that when the probe returns to D, D notices a cross-link and "deletes" either the (A,D) link or the (B,C) link (after a message exchange with B or with C). By symmetry, the cross-links would have been detected by a probe of (A,D) originated by A or a probe of (B,C) originated either by B or C.

Care must be taken in dealing with degenerate crossings caused by exactly collinear links. One way to deal with these is to randomly, but slightly, perturb the reported location of each node to make the likelihood of such links vanishingly small. Another is to carefully define face traversal on the degenerate (zero-area) faces caused by exactly collinear links. To simplify our discussion, we ignore such degeneracies in the rest of the paper.

We have described CLDP in a decentralized fashion, but to understand CLDP's properties, it helps to envision the results of applying CLDP on all links of a static (*i.e.,* unchanging), arbitrary (*i.e.,* no specific connectivity as-

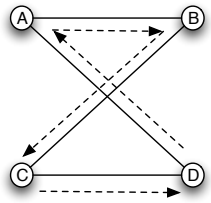

Figure 9: CLDP Probing using right-hand-rule, Case 1.

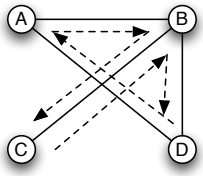

Figure 10: …, Case 2.

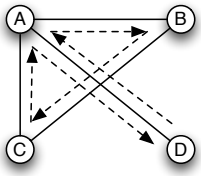

Figure 11: …, Case 3.

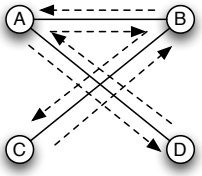

Figure 12: …, Case 4.

sumptions), connected graph. Initially, assume that all the links in this graph are marked *routable*. Then, suppose that each link is probed repeatedly and in some order with the constraint that only one probe is active at any given time (this is an idealization we relax later). As we have described above, a probe may cause a link to be removed. When we say CLDP "removes" a link, we mean that the link is marked *non-routable*. The set of routable links forms a *routable subgraph*. Furthermore, *all CLDP probes traverse the current snapshot of the routable subgraph*. Cross-links are not always marked non-routable; we show later how CLDP preserves cross-links the deletion of which would render the routable subgraph disconnected. The probing stops when subsequent probing of links would not cause any link to be marked non-routable.

As we show in Section 5 (and our simulations of Section 6 bear this out as well), using face routing on the resulting routing subgraph is *guaranteed not to fail* between *any pair of nodes* on *any arbitrary graph*. (We say a routable subgraph is *safe* if face routing is guaranteed not to fail on that subgraph). This result is surprising for the following reason. It is easy to see that CLDP attempts to planarize the routable subgraph by removing cross-links, and face routing is known not to fail on a planarized graph. However, there is no *a priori* reason to believe (and no prior literature that suggests) that using the right-hand rule repeatedly to detect and remove cross-links will always result in a planarization (modulo the cross-links that need to be preserved to avoid disconnections) on an arbitrary graph.

As a practical matter, other forwarding strategies also work perfectly on the CLDP-derived routable subgraphs, such as GPSR's combination of greedy- and perimeter-mode traversals [11], and GOAFR's improvement that uses ellipses to bound face traversals when possible [14]. Note further that greedy forwarding uses the full graph (including links marked "non-routable" by CLDP); only face routing uses the CLDP-derived routable subgraph during recovery from local maxima.

In describing CLDP, we have made two simplifying assumptions: strictly sequential probing of links, and no node or link dynamics. In the following sub-sections we discuss how we augment CLDP to relax these two assumptions. Before doing so, however, we consider two other problems: how CLDP deals with cross-links whose removal would partition the routable subgraph, and how CLDP detects multiple cross-links.

4.2 Partitions in the Routable Subgraph

In Figure 10, the removal of the (B,C) link would disconnect C from the rest of the network. Similarly, in Figure 11, the removal of the (A,D) link would disconnect D, and in Figure 12 the removal of either crossing link would partition the network.

To understand how CLDP deals with this situation, examine the paths taken by the CLDP probes originated by D in each of the figures (by symmetry, one can make similar observations about probes initiated by C). Notice that in every case, when disconnecting a crossing link would partition the graph, the CLDP probe traverses that link *once in each direction*. In Figure 11, for example, the CLDP probe returns to D over the link on which it was sent (*i.e.,* the (A,D) link). Intuitively, it is clear why this should be so: there is no closed face over which the probe can return. In Figure 10, the CLDP probe originated by D traverses link (B,C) once in each direction. From this, B (or C) can infer that removing link (B,C) would cause a partition. Figure 12 shows a scenario where removing either link would partition the routable subgraph.

While we have given the simplest possible examples, our observations generalize easily to arbitrary topologies attached to the "non-removable" link. For example, if in Figure 10, node C were connected to many "clouds" (Figure 13), the CLDP probe would return on the (B,C) link.

Thus, when a CLDP probe traverses either the link being probed (or its cross-link) in both directions, CLDP infers that removal of that link could disconnect the routable subgraph, and does not remove the link. By this rule, CLDP would mark both the (A,D) and the (B,C) links in Figure 12 routable. We point out an important property of the routable subgraphs derived by applying CLDP—*they may contain crossing links*.

Thus, the correct rule for marking links non-routable can be stated as follows. Suppose any node N probes an attached link L and finds a cross-link L':

- Case 1: If both L and L' can be removed (*i.e.,* the CLDP probe traversed neither link twice), remove L.

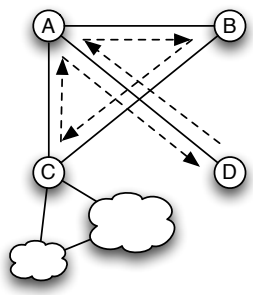

Figure 13: Effect of "clouds" on probes.

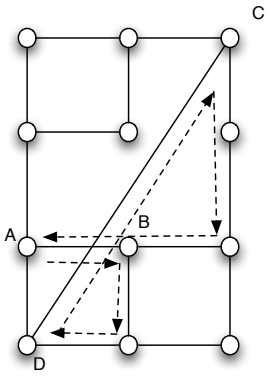

Figure 14: Routable sub-graph depends on probe ordering.

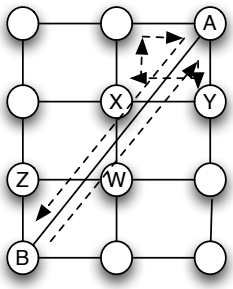

Figure 15: Multiple Cross-Links.

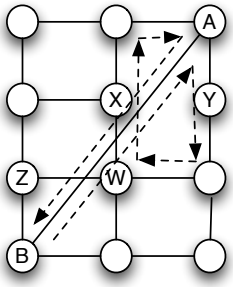

Figure 16: Repeated CLDP probes.

- Case 2: If L can be removed, but L' cannot, remove L.
- Case 3: If L cannot be removed, but L' can, signal the appropriate nodes to remove L'.
- Case 4: If neither link can be removed, do nothing.

Consider the application of this rule to the network in Figure 14, which illuminates. an important property of CLDP: that *different* routable sub-graphs may be generated by applying CLDP to the same graph, depending upon the order in which links are probed. For example, if (A,B) were probed first, then (C,D) would be removed, and vice versa.

4.3 Multiple Cross-Links

Thus far in our discussions, we have assumed that a link is crossed by at most one other link. But consider the situation depicted in Figure 15 where a long link (A,B) is crossed by three other links. In arbitrary graphs, of course, this situation will not be uncommon.

CLDP generalizes rather easily to this case. Repeatedly probing a link until no removable cross-links are found will keep the resulting routable sub-graph safe. Consider Figure 15 and assume that B probes link (A,B). The first such probe will traverse the faces shown, detecting the cross-link (X,Y), which will be removed. A second probe sent by B (Figure 16) will detect the (X,W) cross-link, resulting in the removal of that link (and so on).

Our examples of multiple cross-links are a bit misleading, as they suggest that repeatedly probing a link will detect *all* cross-links. This is not, in general, true: probing *one* of a pair of cross-links is not guaranteed to find the crossing (intuitively, that link may be obscured by other, perhaps non-removable) links. The other link may also have to be probed (from both ends) before the cross-link is detected. Consider, for example, the topology in Figure 17. In this topology, CLDP probes from either end of the (B,C) link are confined to the adjoining triangles, and are unable to detect the (X,Y) link. The (B,C) cross-link is only detected after repeatedly probing the (X,Y) link.

4.4 Concurrent Probing

Thus far, we have assumed that CLDP probes are *serialized*. However, this kind of global serialization is unachievable without significant messaging cost in large networks. A design that permits nodes to probe links concurrently is clearly more desirable.

Unfortunately, concurrent probing can render the routing subgraph disconnected. Consider Figure 9 and assume that while D probes link (A,D), C concurrently probes link (B,C). When each probe returns, C and D each detect a cross-link, and mark their directly attached links non-routable (assume that either link can be removed), leaving the routable subgraph disconnected. Such a race condition can be prevented using a simple *tie-break* rule that deterministically decides which cross-link should be deleted. However, the tie-break rule does not guarantee correctness in the general case.

We now describe a stronger approach called *two phase probing* where a node can removes a link in a face only when no other node attempts to remove links in the same face. In this approach, CLDP cross-link detection and removal is split into two phases. In the first "probe" phase, a *probe* message for each link traverses a face to see if the link is crossed by other links in the face. When the *probe* message has returned to sender node, if its result is case 1 or case 2 as described in Section 4.2, the node initiates a second "commit" phase. If its result is case 3, the node signals the other node, which has a cross link detected in "probe" phase, to initiate "commit" phase. In the second "commit" phase, a node sends a *commit* message to the probed link and sets link state to "committing". A *com-*

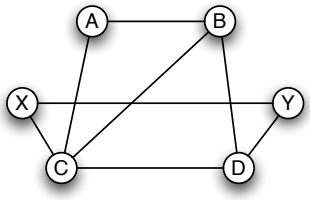

Figure 17: Probing a link may not detect a cross-link.

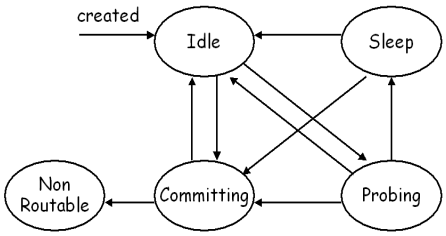

Figure 18: Link state diagram for CLDP.

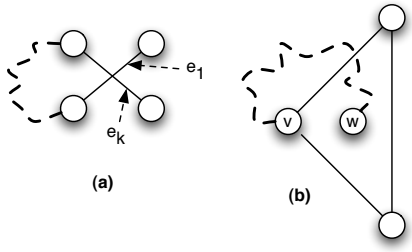

Figure 19: Figures for proof sketches

mit message informs all nodes in the traversing face that the sender of the *commit* will remove a link from its planarized graph. Furthermore, if multiple committers concurrently walk in the same face, a *commit* message is used to determine exactly one winner. CLDP uses a simple tie-break to do this: when a *commit* message reaches another "Committing" link in the traversing face, if its link identifier is less than link identifier in the *commit* message, the *commit* message is dropped. Conversely, if its link identifier is greater, the *commit* message continues its face walk and the link aborts its commit phase.

The state diagram in Figure 18 describes two phase probing. Each node participating in the procedure is in one of five states:

Idle : The link is periodically and continuously probed.

Probing : After a *probe* message is sent, the link keeps "Probing" state until the *probe* message has returned or a probing timer is expired.

Sleep : If the result of the *probe* message is "no cross link" or case 4 described in Section 4.2, the link set its state to "Sleep". Moreover, in case of the latter result, the link is marked with "keep-link" flag. A "sleep"-ing link can be woken up later. [7].

Committing : After the link sends a *commit* message, it keeps "Committing" state until the *commit* message has returned, or another *commit* message that overrides its own (using the commit tie-break described above) is observed, or committing timer is expired. If a *probe* message or a *signal* is observed in "Committing" links, it is dropped.

Non-routable : If a *commit* message is successful, the committed link is removed from planarized graph. When this happens, the two adjacent links obtained by applying both right-handed rule and left-handed rule to "Non-routable" link set their states to "Idle" and set their "wakeup" flags.

[7]When a *commit* message with a special "wakeup" flag is observed, an adjacent link is removed from planarized graph, or an adjacent neighbor is disconnected

Our focus in this paper has been to demonstrate the existence of one mechanism that renders CLDP race-free. Accordingly, we have implemented this described mechanism, and our simulations suggest that it works well.

4.5 Link Addition and Deletion

To make CLDP practical, we must augment it to behave correctly under dynamics, such as node and link failures.[8] In Section 6, CLDP with the *two phase probing* mechanism described in Section 4.2 works well under network dynamics, such as link addition and deletion.

4.6 Putting It All Together

In the preceding sub-sections, we have described various aspects of CLDP's design. In our discussions, we have assumed arbitrary connectivity; this is a radical departure from models considered in previous geographic routing proposals, which are largely based on unit-graph assumptions or variants thereof. *Why* does CLDP result in safe routable subgraphs? Essentially, CLDP is a distributed planarization procedure that finds cross-links in a graph, and eliminates them when doing so would not disconnect the routable subgraph. Face routing on this "almost" planar graph never fails. This is the intuition that we make more rigorous in the next section.

5 Proof Sketches of Correctness

Because of space constraints, we provide sketches of proofs for our two basic results. Recall that, as stated in Section 4.1, we assume that full graphs have no degeneracies: no vertices are coincident, and no pairs of edges at a single node have the same incident bearing. We use the following notation. The set of edges of a graph is denoted by E, and individual directed edges are denoted by e_i, with

[8]We have not explicitly considered node mobility in our evaluations of CLDP, but have left that to future work. In principle, CLDP wouldn't need additional mechanisms to function under mobility, and would work well when link disconnections due to mobility occur on much longer timescales than the time required to complete CLDP probes.

e_{-i} denoting an edge in the opposite direction. Since we have assumed symmetrical links, $e_i \in E \Rightarrow e_{-i} \in E$. The set of vertices is denoted by V, and the starting point of an edge is given by $s(e_i) \in V$ and the finishing point of an edge is given by $f(e_i) \in V$. A path is a sequence of edges, $e_1, e_2, \ldots$ such that $s(e_{i+1}) = f(e_i)$. For each graph define a (perhaps empty) set of crossings C; each element of C is a pair of edges that intersect in the plane.

Note that all face walks eventually return to their starting points. We use the following terminology to describe how they return. An edge is *singly-walked* if a face walk starting on that edge does not return via that same edge (in the opposite direction). An edge is *doubly-walked* if it returns via the same edge in the opposite direction. The general rule in CLDP is that when a crossing is detected, no doubly-walked edge can be removed.

We start with a general observation about crossings in connected graphs.

Theorem 5.1 *If a connected graph G has at least one crossing, then there is at least one face that has a crossing.*

Sketch of Proof: Consider a connected graph G that has at least one crossing; *i.e.*, C is nonempty. Then there is some pair of crossed edges, call them e_1, e_k, and a path between these crossed edges that we denote by $e_1, e_2, \ldots e_k$. If there are any pairs of edges on this path that are in C, then we can choose that crossing instead (using the subset of the path between these two crossed edges). Repeating this, we find a crossing and a path such that the path contains no other crossings. We then have a situation as in Figure 19(a).

The portion of the path between the crossing point and around the series of edges back to the crossing point has a well-defined interior and exterior. Among all such configurations like that in Figure 19(a), we pick the one with the minimal area in the interior.

We now start a face walk at edge e_1 (we can assume, without loss of generality, that the right-hand rule from e_1 points towards the interior of the path; if not, we start the walk at e_k). We know the face walk must eventually return to $s(e_1)$. Thus, the face walk must eventually cross the path, because the point $s(e_1)$ is exterior to the path and the face walk is oriented inwards (so any deviations from the path point inwards). If the face walk passes through edge e_k, or crosses itself, we are done. If the face walk does not pass through e_k and does not cross itself then it must (a) leave the path at some point, call it v, and (b) cross the path somewhere other than at e_k (and farther along the path than v). Let e_j denote the link that crosses the path, and e_r be the link that it crosses. Then we have a new crossing pair, e_j, e_r, with a path that is comprised of the old path from v to e_r and the face walk from v to e_j. This path outlines a strict subset of the area outlined by the previous path. This contradicts our minimality assumption. Thus, the face walk for this minimal area path must cross itself. *QED.*

This result shows that if we had used a version of CLDP that eliminated *all* crossings then we would end up with a set of connected planar components.

To help state our next result, we term a graph *CLDP-stable* if CLDP would not eliminate any edge in the graph, were the edges probed in serial fashion.

Theorem 5.2 *Geographic routing never fails on a connected CLDP-stable graph.*

Sketch of Proof: Assume that a connected graph G is CLDP-stable but has a routing failure. As we discussed in Section 2, a routing failure from a point v to a point w occurs when a face walk starting on a link that straddles the line segment between v and w does not pass through the $\overline{vw}$ line segment at a point closer than where the originating link crosses. Thus, the face walk must *only* cross $\overline{vw}$ either at or behind the previous crossing; or through the continuation of $\overline{vw}$ behind v; or through the continuation of $\overline{vw}$ beyond w. Here we assume the latter case, but our discussion with slight modification applies to any of the three cases. For this case, we have a picture that looks generally like that shown in Figure 19(b).

We have a face that surrounds the point w, and a path from v to w that intersects the face; let edge e_k denote the edge that intersects this face and e_1 denote the edge on the face that is crossed. We have a path $e_1, e_2, \ldots, e_k$ from e_1 back to the crossing at e_k. All edges on the face surrounding w are singly-walked, so the crossing involves at least one singly-walked edge. We proceed to prove that such an example violates CLDP-stability; in other words, the presence of a crossing with at least one singly-walked edge implies that there is a face walk with a crossing with at least one singly-walked edge.

We can't, as above, insist that there are no other crossings on the path. However, we can choose an example where all such intervening crossings involve two doubly-walked edges; by the same reasoning as above, if there were another crossing involving at least one singly-walked edge then we could pick that crossing instead. Thus, we have a picture like that in Figure 19(a) except that here the path can have self-intersections.

We now construct a series of subpaths from this path that will have a well-defined interior and exterior. Consider the set $\tilde{C}$ of crossings for which both edges are in the path. Define the *class-1 boundary* as follows. Starting at edge e_1, continue along the path until you hit an edge e_j with a pair $(e_j, e_r) \in \tilde{C}$ for some e_r; at this point stop the subpath and jump to the last occurrence of e_r in the path, and continue until another crossing is encountered. Repeat this process until you reach e_k. This results in a series of paths which intersect at the end edges; this boundary begins at edge e_1 and ends at edge e_k. This boundary has a well-defined interior and exterior.

One can define the set of *class-2 boundaries* by starting at each of the crossings and following the original path rather than the shortcut (in this case, one such subpath

would start at e_j and end at e_r). Similarly, one can recursively define class-z boundaries for increasing values of z. Of the set of all paths connecting crossings that have at least one singly-walked link, pick one with the minimal class-1 area. If there are several of those, pick the one with the minimal class-2 area, *etc.*

We now define a *reduced face walk*. A reduced face walk eliminates the portion of the face walk in between walks over a doubly-walked edge. That is, when a face walk encounters a doubly-walked edge e_i, it will go on some path and then return via e_{-i}. The reduced face walk eliminates the portion of the path $e_i, \ldots, e_{-i}$. The resulting reduced face walk is the same as if the doubly-walked links did not exist. Note that a reduced face walk consists solely of singly-walked edges.

We now apply the reasoning from the previous proof, except we use reduced face walks, and we are concerned only about crossings of the boundaries, not the path itself (the path might enter the interior of the boundary). As before, we start a reduced face walk with the edge that we know to be singly-walked. If the face walk passes through e_k or crosses itself we are done. If not, then we follow the reduced face walk until we cross the boundary. At this point we have an example of a crossing connected by a boundary that sweeps out a smaller area. Moreover, because we used the reduced face walk, we know that the crossing involves at least one singly-walked link. This line of reasoning needs to be adjusted somewhat based on where the first deviation from the path occurs. If it occurs before the first path crossing, then we use the class-1 boundary; if it occurs after the first crossing, we use the class-2 boundary, *etc.* In this way, we always are doing a reduced face walk within a boundary of minimal area and find a contradiction when we cause a new crossing. *QED.*

6 Simulation Results

Our proof focused on CLDP's correctness on static graphs. However, to show that CLDP is practical on real wireless networks, we examine the performance of CLDP through simulation.

Methodology and Metrics We implemented CLDP (and other geographic routing protocols, described below) in TinyOS [8], the event-driven operating system used on the Mica-2 motes. TinyOS code can be directly executed on TOSSIM [16], a process-level simulator that can be used to directly debug and evaluate sensor network applications and protocols. In this section, we report simulation results obtained from running CLDP and other protocols using TOSSIM's support for packet-level simulation.

Our implementation of CLDP in TinyOS is 750 lines of nesC code. This implementation largely follows the description of the protocol in Section 4. GPSR is used to route packets to their destinations.

In this section, we compare (whenever appropriate) CLDP's performance against three alternatives, *GPSR* denotes the full implementation of GPSR using the Gabriel Graph for planarization, greedy forwarding, and perimeter traversal for routing around voids. We use *GPSR* to provide context for CLDP's performance. $GPSR-PLAN$ denotes a protocol that forwards packets using GPSR on the full connectivity graph (*i.e., without* planarization). $GPSR-PLAN$ delineates the baseline performance of face walking on the networks we study. $GPSR+MWP$ includes, in addition to GPSR and planarization, an implementation of the "mutual witness" procedure for avoiding unidirectional links and disconnections in the planarized graph whenever the unit-graph assumptions are violated (Section 3). $GPSR+MWP$ quantifies the inadequacy of that proposed fix for planarization failures, thereby highlighting the need for CLDP.

In each of our simulations, we use a 200-node topology in which nodes are randomly positioned on a fixed-size two-dimensional surface. We conduct simulations on two types of networks: wireless networks with an idealized radio model with circular radio ranges (we introduce reality in the form of obstacles), and Bernoulli random graphs which have a fixed connection probability for any pair of nodes, regardless of Euclidean distance between the nodes. For our wireless network simulations, we evaluate the performance of our various geographic routing protocols as a function of node density. Our measure of density is the average number of neighbors of a node. We scale the area of the surface in order to vary node density; for our highest density we use an area of 1300 by 1300 units, while for our lowest, we use an area of 2000 by 2000 units. The radio range is 180 units.

In our simulations with obstacles, the number of obstacles is indicated by a parameter f, such that fN is the total number of obstacles (N is the number of nodes). Each obstacle is of fixed length (45 units) in each of our simulations. The mid-point of the obstacle is randomly positioned on the two-dimensional surface, and the orientation of the obstacle is equally likely to be either vertical or horizontal. This obstacle model helps us stress CLDP and other protocols to varying extents in order to measure their performance.

Our Bernoulli random graphs are generated in the obvious way: we flip a coin for each pair of nodes, assigning a link between them with the desired connection probability.

For each simulation run we first generate a network topology. We then ensure that the topology is connected. At the beginning of the simulation, TOSSIM enforces a boot-up time during which nodes are started randomly. In our simulations, our 200 nodes are started randomly in the first 30 seconds. Following the boot phase, each simulation run consists of two phases. In the first phase, we let the appropriate routability determination protocol (CLDP, or GPSR's planarization and/or mutual witness

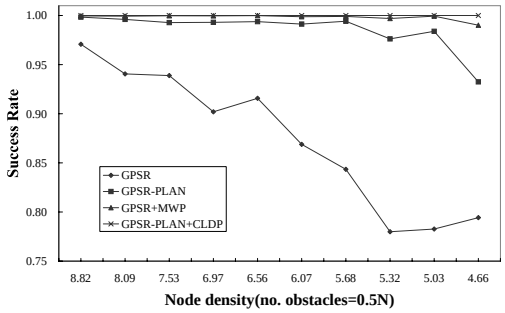

Figure 20: Success rate for $0.5N$ obstacles.

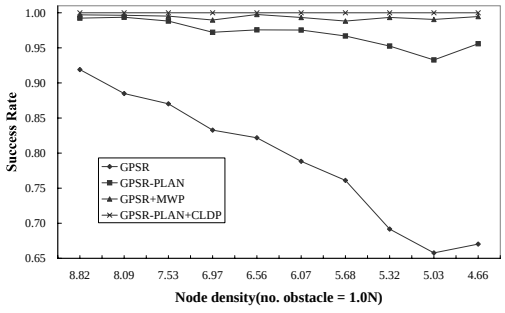

Figure 21: Success rate for $1.0N$ obstacles.

procedure) execute at each node long enough for the network to converge. In the second phase, we send packets pairwise bidirectionally between nodes in a staggered manner to minimize wireless collisions. This latter phase tests for routing failures. For each data point in the graphs below, we run 50 random topologies. We have verified that this is sufficient to produce negligible 95% confidence intervals for the average values of our metrics.

We do not simulate packet losses due to interference or buffer overrun in either phase. Packet losses would increase the convergence time of CLDP, or would alter the level of concurrent probing in CLDP. Our simulation methodology already introduces significant concurrency by ensuring that all nodes start at nearly the same time. More detailed simulations with realistic loss models is left to future work. Our simulations do drop packets, however, when face routing fails.

We use two primary measures of performance. The *success rate* metric measures the fraction of sender/receiver pairs for which packet transmission from a sender is successfully received. The *average stretch* measures the average of path stretch for each sender/receiver pair. The stretch of a path is the ratio of the number of hops using the appropriate routing scheme to the number of hops in the shortest path. We also evaluate the overhead and convergence time of CLDP; we define these metrics below.

Wireless Networks with Obstacles Figures 20 and 21 show the success rate as a function of node density for our various protocols when the number of obstacles is $0.5N$ and $1.0N$ respectively. As expected, CLDP allows perfect delivery success across all node densities we evaluated. Interestingly, GPSR's planarization procedure fails rather dramatically in the presence of even a moderate number of obstacles. In these circumstances, it appears to be more advantageous to simply use GPSR on the connectivity graph without planarization. The mutual-witness procedure fixes many of GPSR's shortcomings and is close to being perfect in some cases. With fewer obstacles it achieves 100% success at all but two values for density, but with more obstacles it is never perfect except at the highest density. As we argue in footnote 2, we believe it is unacceptable for a routing protocol to exhibit persistent routing failures, even if for only a few source-destination pairs.

Figures 23 and 24 plot the average stretch as a function of node density for our various protocols when the number of obstacles is $0.5N$ and $1.0N$ respectively. CLDP exhibits an average stretch between 2 and slightly above 4, with a higher stretch at lower densities and when there are more obstacles. CLDP outperforms GPSR+MWP in this respect. CLDP removes only cross links. In contrast, GPSR+MWP removes all links that are witnessed by planarization. Hence, this property makes CLDP to get better performance than GPSR+MWP. Because GPSR succeeds in "easy" paths and fails, intuitively, "difficult" paths for which CLDP and MWP have to "work" harder (*i.e.*, have longer path lengths), GPSR is not shown in these figures that are results from computing stretch only for successful paths. This is evident from the CDF of stretch for CLDP (Figure 22, with $1.0N$ obstacles). Notice the long tail of the distribution with some paths having a stretch of over 100! However, across the range of densities we explore, 60-95% of the paths have a stretch less than 2.

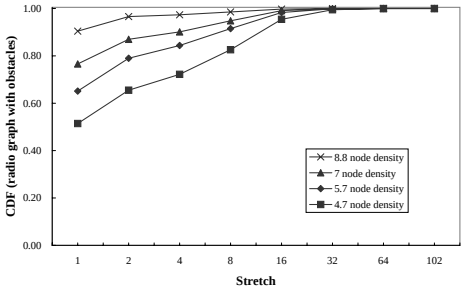

Figure 22: CDF of stretch ($1.0N$ obstacles).

Finally, we have computed our two metrics for two lev-

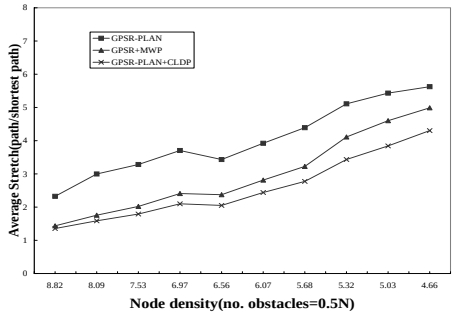

Figure 23: Average stretch for $0.5N$ obstacles.

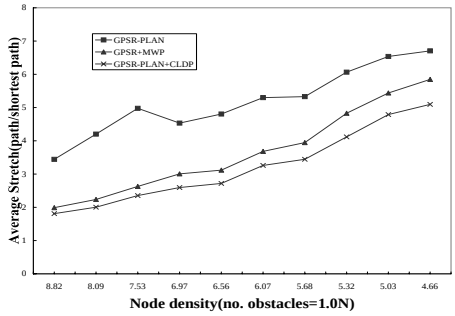

Figure 24: Average stretch for $1.0N$ obstacles.

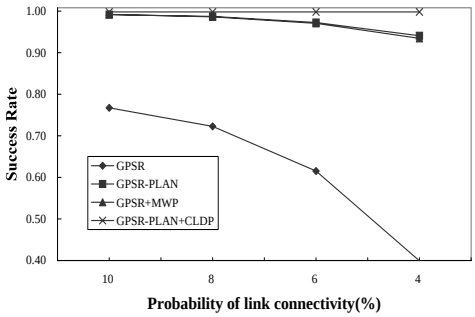

Figure 25: Random graph success rate.

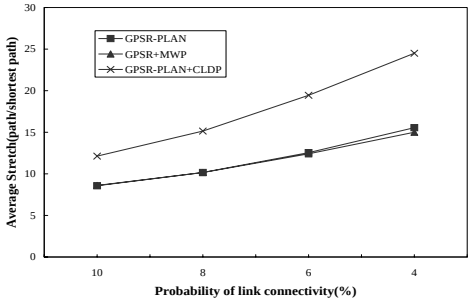

Figure 26: Random graph average stretch.

els of localization error, when the error in each coordinate is uniformly distributed about an interval of width $0.15R$ and $0.30R$ respectively, where R is the nominal radio range. The behavior of each of our protocols is qualitatively similar to that with obstacles, so we omit those graphs for brevity.

Random Graphs To stress CLDP, we also simulated it on Bernoulli random graphs with various connectivity probabilities. As Figure 25 shows, CLDP exhibits no routing failures even on random graphs. By contrast, all other variants exhibit significant routing failures on sparse random graphs (low connection probabilities). In particular, MWP exhibits more systematic routing failures than on wireless networks. Clearly, none of these protocols are practical for routing on random graphs.

As one would expect, CLDP's stretch is rather high for the reason described above. On some paths, CLDP exhibits a stretch above 400 (graph not included for space reasons). GPSR's stretch is not compared with CLDP'stretch since GPSR fails most paths.

Overhead We measured how many CLDP messages are needed to add a link to wireless networks with $1.0N$ obstacles and to Bernoulli random graphs. This gives us some idea of the overhead incurred by CLDP. In our experiments for measuring overhead, after a network has reached steady state, two nodes not directly connected to each other are randomly selected and an additional link between two nodes is activated. The *overhead* is the total number of CLDP control messages (probe and commit) traversing a link in either direction until network has converged.

Figure 27 plots the distribution of link overheads averaged over 20 times link-addition on each 50 wireless topologies. It shows that about 85%-90% of links see less than 4 messages, but a very small fraction of links see upwards of 100 messages. This latter phenomenon can be explained as follows. Assume that a new link is added which crosses existing edges. When CLDP removes these crossing edges, it needs to wake up all links on the faces adjacent to the removed link in order to detect successively hidden cross-edges. These links generate *probe* messages to see if they are crossed by others. Hence, the number of message observed in a link depends on the size of the face. Clearly, in our wireless topologies (particular in the ones with lower density), there exist long faces. This effect is more pronounced for our random topologies. Figure 28 plots the distribution of link overheads averaged over 20 runs on 50 Bernoulli random topologies. It shows that about half of links see less than 2 messages and about

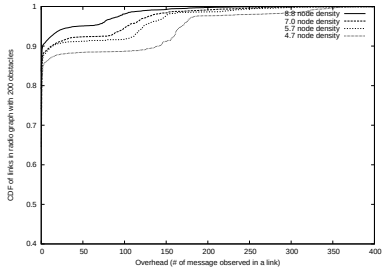

Figure 27: Overhead for wireless network with 1.0N obstacles.

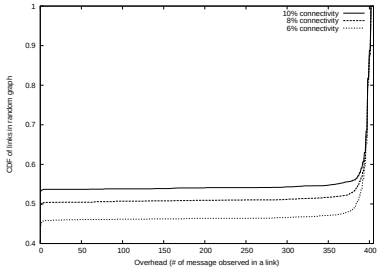

Figure 28: Overhead for Bernoulli random graphs.

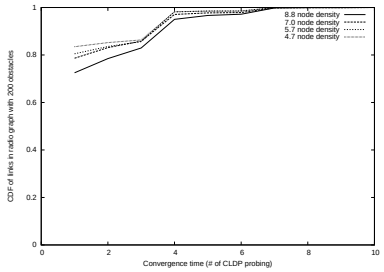

Figure 29: Convergence time distributions for wireless network with 1.0N obstacles.

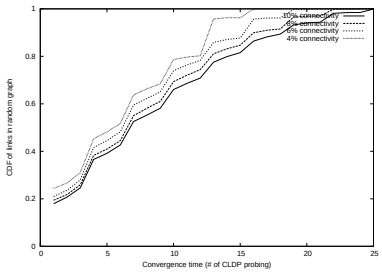

Figure 30: Convergence time distributions for Bernoulli random graphs.

half of links see between 380 and 400 messages.

Network Convergence Time We measured how fast CLDP converge both on wireless networks with 1.0N obstacles and on Bernoulli random graphs. In experiments of convergence time, 200 nodes are initially started roughly simultaneously. The *convergence time* of a link is defined as the time after which link's state becomes "Sleep" or "Non-routable" (*i.e.*, a routable link remains routable, a non-routable link remains non-routable). Figure 29 and 30 show the convergence time distribution for various links on wireless networks and Bernoulli random graphs. The convergence time is the number of probing to links. In Figure 29, about 95% links converge within 4 times link probing intervals and all links converge within 9 such intervals. For random graphs (Figure 30), as expected, link convergence times are longer. Even so, all links in our random graphs converge within 25 probing intervals.

Network Dynamics Finally, we conducted experiments to evaluate how resilent CLDP is to network dynamics. These experiments were done on 200 wireless networks with 1.0N obstacles as well as 200 Bernoulli random graphs. In all experiments, we took each given topology, and randomly selected some links and marked them as "Non-routable". Then, we let CLDP execute at each node. Initially, these "Non-routable" links are not used for CLDP probing. Over time, these links are woken up, eventually become "Idle", and are CLDP-probed. After probings to all links are stopped, we sent packets between each pair of nodes in staggered manner and to check if they are successfully delivered to destination node.

Node density	8.8	7.0	5.7	4.7
Success rate	100%	100%	100%	100%

Table 2: Result of network dynamics experiments on wireless networks.

Connection prob.	10%	8%	6%	4%
Success rate	100%	100%	100%	100%

Table 3: Result of network dynamics experiments on Bernoulli random graphs.

As is shown in Table 2 and 3, CLDP on Bernoulli random graphs as well as wireless networks with 1.0N obstacles achieve 100% success at network dynamics experiments.

Summary In all of our simulation experiments, we found not *one* counter-example for CLDP's correctness. CLDP exhibits reasonable stretch, overhead, and convergence times. as well as it work well under network dynamics. There is also room to examine techniques for lower overhead.

7 Conclusion

In this paper we have motivated, described, and evaluated CLDP, which, to our knowledge, is the first distributed planarization protocol that renders geographic routing provably correct on arbitrary graphs. While our initial simulations indicate that CLDP is quite practical, significantly more evaluation and experimentation is required to fine-tune some of CLDP's mechanisms. We are right now poised to deploy CLDP on a testbed of 20 Mica-2 motes, using the same nesC implementation used for the simulations in this paper. Our next experiments on our testbed, will focus on observing CLDP's behavior under network dynamics, where nodes and links fail frequently.

We close by observing that we expect CLDP will be of great benefit for performing geographic routing without location information. While previous work [19, 18] has had to assign nodes coordinates in a way that *both* ensures routing correctness *and* offers minimal stretch, CLDP decouples coordinate assignment from routing correctness, as it renders geographic routing successful on arbitrary graphs. We therefore expect that it will be straightforward to develop efficient coordinate assignment techniques for use in conjunction with CLDP in wireless networks without localization capabilities that produce routes with low stretch.

References

[1] BISWAS, S., AND MORRIS, R. Opportunistic routing in multi-hop wireless networks. In *Proceedings of the ACM Workshop on Hot Topics in Networks* (Boston, MA, USA, Nov. 2003).

[2] BOSE, P., MORIN, P., STOJMENOVIC, I., AND URRUTIA, J. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications* (Seattle, WA, USA, Aug. 1999), ACM, pp. 48–55.

[3] FINN, G. Routing and addressing problems in large metropolitan-scale internetworks. Tech. Rep. ISI/RR-87-180, USC/Information Sciences Institute, Mar. 1987.

[4] GABRIEL, K., AND SOKAL, R. A new statistical approach to geographic variation analysis. *Systematic Zoology 18* (1969), 259–278.

[5] GANESAN, D., ESTRIN, D., WOO, A., CULLER, D., KRISHNAMACHARI, B., AND WICKER, S. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Tech. Rep. UCLA/CSD-TR-02-0013, University of California, Los Angeles, Computer Science Department, 2002.

[6] GAO, J., GUIBAS, L., HERSHBERGER, J., ZHANG, L., AND ZHU, A. Geometric spanner for routing in mobile networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Oct. 2001), pp. 45–55.

[7] GIORDANO, S., STOJMENOVIC, I., AND BLAZEVIC, L. *Position-based Routing Algorithms for Ad-Hoc Networks: A Taxonomy*. Kluwer Publishers, 2003.

[8] HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems* (Cambridge, MA, USA, Nov. 2000), ACM, pp. 93–104.

[9] KARP, B. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.

[10] KARP, B. Challenges in geographic routing: Sparse networks, obstacles, and traffic provisioning. Presentation at the DIMACS Workshop on Pervasive Networking, May 2001.

[11] KARP, B., AND KUNG, H. T. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* (Boston, Mass., USA, Aug. 2000), ACM, pp. 243–254.

[12] KLEINROCK, L., AND TAKAGI, H. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications 32*, 3 (1984), 246–257.

[13] KO, Y.-B., AND VAIDYA, N. Location-aided routing in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* (Aug. 1998).

[14] KUHN, F., WATTENHOFER, R., ZHANG, Y., AND ZOLLINGER, A. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing* (Boston, MA, USA, July 2003).

[15] KUHN, F., WATTENHOFER, R., AND ZOLLINGER, A. Ad-hoc networks beyond unit disk graphs. In *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing* (Sept. 2003).

[16] LEVIS, P., LEE, N., WELSH, M., AND CULLER, D. TOSSIM: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the ACM Sensys* (2003), ACM Press, pp. 126–137.

[17] LI, X., KIM, Y. J., GOVINDAN, R., AND HONG, W. Multi-dimensional range queries in sensor networks. In *Proceedings of the ACM Sensys* (Los Angeles, CA, USA, Nov. 2003).

[18] NEWSOME, J., AND SONG, D. GEM: Graph embedding for routing and data-centric stroage in sensor networks with geographic information. In *Proceedings of the ACM Sensys* (Nov. 2003).

[19] RAO, A., RATNASAMY, S., SHENKER, S., AND STOICA, I. Geographic routing without location information. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* (Oct. 2003), pp. 96–108.

[20] RATNASAMY, S., KARP, B., YIN, L., YU, F., ESTRIN, D., GOVINDAN, R., AND SHENKER, S. GHT: A geographic hash table for data-centric storage. In *Proceedings of the ACM Workshop on Sensor Networks and Applications* (Atlanta, Georgia, USA, Sept. 2002), ACM, pp. 78–87.

[21] SAVVIDES, A., HAN, C.-C., AND SRIVASTAVA, M. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* (Rome, Italy, July 2001), ACM, p. to appear.

[22] SEADA, K., HELMY, A., AND GOVINDAN, R. Localization errors on geographic face routing in sensor networks. In *Proceedings of the IEEE International Workshop on Information Processing in Sensor Networks* (Berkeley, CA, USA, Apr. 2004).

[23] SOKAL, R., AND MATULA, D. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis 12* (1980), 205–222.

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

[25] WOO, A., TONG, T., AND CULLER, D. Taming the underlying challenges of reliable multihop routing. In *Proceedings of the ACM Sensys* (Los Angeles, CA, November 2003).

[26] ZHAO, J., AND GOVINDAN, R. Computing aggregates for monitoring wireless sensor networks. In *Proceedings of the International Workshop on Sensor Net Protocols and Applications* (Anchorage, AK, April 2003).

[27] ZHAO, J., AND GOVINDAN, R. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the ACM Sensys* (Los Angeles, CA, November 2003).