Robust Position-Based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges

[Extended Abstract]

Lali Barrière
Departament de Matemàtica Aplicada i Telemàtica
Univ. Politècnica de Catalunya
Barcelona, Spain
lali@mat.upc.es

Pierre Fraigniaud
CNRS, Laboratoire de Recherche en Informatique
Université Paris-Sud
Orsay, France
pierre@lri.fr

Lata Narayanan and Jaroslav Opatrny
Department of Computer Science
Concordia University
Montreal, Canada
lata@cs.concordia.ca

ABSTRACT

Several papers showed how to perform routing in *ad hoc* wireless networks based on the positions of the mobile hosts. However, all these protocols are likely to fail if the transmission ranges of the mobile hosts vary due to natural or man-made obstacles or weather conditions. These protocols may fail because in routing either some connections are not considered which effectively results in disconnecting the network, or the use of some connections causes livelocks. In this paper, we describe a robust routing protocol that tolerates up to roughly 40% of variation in the transmission ranges of the mobile hosts. More precisely, our protocol guarantees message delivery in a connected adhoc network whenever the ratio of the maximum transmission range to the minimum transmission range is at most $\sqrt{2}$.

General Terms

Mobile computing and Communications

Keywords

Wireless Networks, Ad Hoc Networks, Routing

1. INTRODUCTION

An *ad hoc* network is a network consisting of mobile hosts that is established as needed, not necessarily with any assistance from the existing internet architecture. The mobile hosts can communicate with each other using wireless broadcasts. However, we allow the possibility that not all hosts are within the transmission range of each other. Thus, communication between two hosts is achieved by multi-hop routing, where intermediate nodes cooperate by forwarding packets. Host mobility means that the topology of the network can change with time. Furthermore, no assumption can be made about the initial topology of the network. As a consequence, nodes have to build and update their routing tables automatically . In the last few years, a large number of routing protocols have been proposed for mobile ad hoc networks [5, 15, 16, 14]. Most of these use limited *flooding* of control packets to obtain topology information. However, flooding can use up significant amounts of scarce wireless bandwidth.

Recently, many authors have proposed the use of *location information* to reduce the amount of control traffic [1, 3, 8, 9]. Every node is assumed to know its own location, for example, by using the global positioning system (GPS). Ko and Vaidya [9] essentially use the DSR protocol [5], but suggest that a node forward a packet to a node only if it is in a *request zone* which is likely to contain a path to the desired destination. The DREAM protocol [1] uses a limited flooding of data packets. On the other hand, *greedy routing* [12] and *compass routing* [11] completely stay away from the flooding paradigm, and at every step, a node forwards a packet to exactly one of its neighbors. This neighbor is selected by locally optimizing the length of the path to the destination.

The standard model for wireless ad hoc networks represents the mobile hosts spread out in some environment by a set of points in the Euclidean plane. The transmission range R specifies the maximum (Euclidean) distance between two mobile hosts at which they are able to directly communicate. A mobile ad hoc network can thus be represented by a *unit disk* graph: given an embedding of nodes in the plane, two nodes are adjacent if and only if they are at distance at most R. We assume that each node is aware of its own position in the plane, that is, it learns its x- and y-coordinates, for example, by using GPS. Mobile hosts are henceforth identified by their coordinates in the plane.

When a mobile host u wants to send a packet to another host v located at distance greater than R from u, other mobile hosts are required to relay the packet. *Greedy* routing, also called *geographic distance routing* [12] constructs a sequence of mobile hosts $(w_0, w_1, \ldots, w_k)$ where $w_0 = u$ and $w_k = v$, and w_{i+1} is selected by w_i as its neighbor closest to

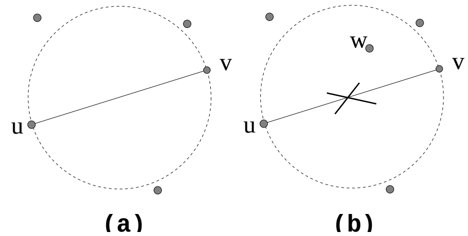

Figure 1: (a) The edge $\{u, v\}$ is kept. (b) The edge $\{u, v\}$ is removed.

the destination v. Obviously, there are several scenarios in which greedy routing fails. For instance, if the mobile host w_i is closer to the destination than any of its neighbors, or if two adjacent mobile hosts w_{i-1} and w_i are equally close to the destination, and none of their other neighbors is closer, then a livelock results.

An alternative to greedy routing is *compass routing* [11], also called *directional routing*, for which several improved variants have been proposed in [1, 10]. In compass routing, the next node w_{i+1} is chosen among w_i's neighbors to minimize the angle between $w_i v$ and $w_i w_{i+1}$. Unfortunately, there are again known scenarios in which compass routing fails. Combining compass and greedy routing fails as well, although there are large classes of configurations for which this combination of the two routing strategies ensures message delivery [2, 3].

To overcome the failure of all these routing strategies, Bose *et al.* [4] described a routing algorithm which guarantees delivery of messages in ad hoc wireless networks. (The transformation of this algorithm into a protocol named GPSR was later presented by Karp and Kung [8].) In order to ensure delivery, the algorithm uses *perimeter routing* in the so-called *Gabriel graph* of the network [6]. The Gabriel graph is a spanning subgraph of the original network. It is defined as follows: given any two adjacent nodes u and v in the network (that is, two nodes at distance at most R), the edge $\{u, v\}$ belongs to the Gabriel graph if and only if no other node w of the network is located in the disk with the segment (u, v) as its diameter (see Figure 1). It is known that the Gabriel graph is planar and connected if the underlying graph is a unit disk graph [13]. Once the Gabriel graph is extracted from the network, routing is performed along its edges. Its planarity and its connectivity ensure message delivery by routing along the faces of the graph (e.g., by using the right-hand-rule) that intersect the line between the source and the destination. A rough idea of perimeter routing is given in Figure 2.

To improve the efficiency of the algorithm in terms of routing performance, perimeter routing can be used in combination with greedy routing [8]. Routing is mainly greedy, but if a mobile host fails to find a neighbor closer than itself to the destination, it switches the message from "greedy" state to "perimeter" state. A message in the perimeter state is routed around the faces of the Gabriel graph. As far as the message delivery is concerned, one could leave the message in this state until it reaches its destination. However, it

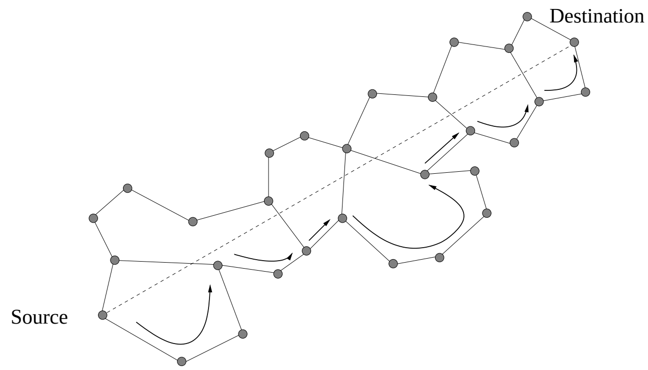

Figure 2: Perimeter routing.

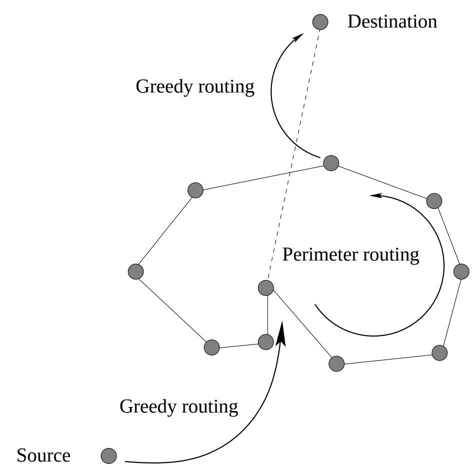

Figure 3: Rough description of alternating greedy and perimeter routing.

is more efficient to move the message back to the greedy state once it has reached the "other extremity" of the face, if possible (see Figure 3).

Although the algorithm in [4] is the first known distributed protocol ensuring message delivery in ad hoc networks without the use of flooding [9, 12], it is likely to fail if there is some instability in the transmission ranges of the mobile host. Instability in the transmission range means that the area a mobile host can reach is not necessarily a disk and the range can vary between $r = (1 - \epsilon)R$ and R, $\epsilon > 0$ as shown in Figure 4.

In such cases of instability, two mobile hosts at distance d, where $r < d \leq R$, may or may not be able to communicate directly. This situation commonly occurs if there are obstacles (for example, buildings, mountains, etc.) between the mobile hosts, which disrupt the radio transmissions. Perturbations can also be caused by bad weather conditions (heavy rain, snow storm, etc.) or other unrelated radio transmissions. Even a slight variation of the ranges with ϵ arbitrarily small can cause subtle problems. For example, the variation in transmission range may create some unidirectional communication links which would not exist if all hosts had the

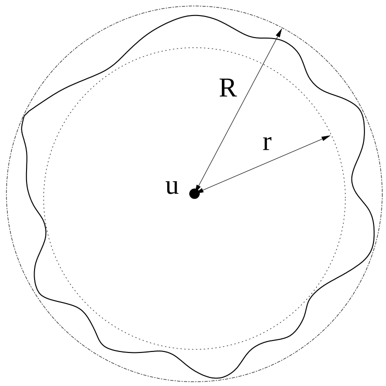

Figure 4: The transmission range of the mobile host u varies between r and R.

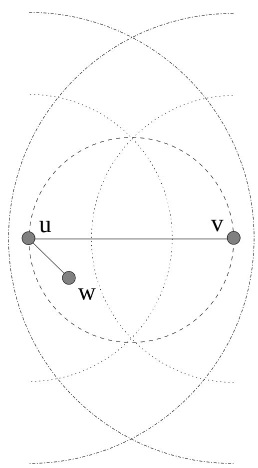

Figure 5: Failure of the Gabriel graph construction. Nodes u and v do not agree about whether or not to keep the edge $\{u, v\}$.

same uniform transmission range. Bidirectional communication links can be enforced in a network by requiring that a communication link from the host u to the host v is valid only when an acknowledgment from v is received by u. Even when only bidirectional communications are assumed, the protocols in [4, 8] may fail due to a slight variation of the ranges. The main reason is that the Gabriel graph which is used for routing is not a robust structure. For instance, Figure 5 shows a case in which the vertices u and v do not agree on the existence of the edge $\{u, v\}$ in the corresponding Gabriel graph. According to u, the host w is in the disk around the segment (u, v) and thus the edge $\{u, v\}$ should not be in the Gabriel graph. However, the host v does not see w in the disk and would retain the edge $\{u, v\}$ in the Gabriel graph. Figure 6 shows that it is not easy to overcome such a disagreement because removing the edge may disconnect the network, but keeping the edge may create a non-planar graph and the planarity of the Gabriel graph is crucial for the perimeter routing.

In this paper, we present a distributed routing protocol

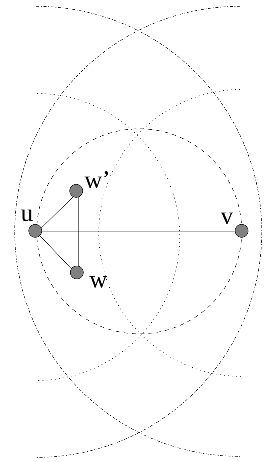

Figure 6: Removing the edge $\{u, v\}$ results in a disconnected graph while retaining it results in a non-planar graph.

which ensures the delivery of messages even if transmission ranges fluctuate by roughly 40% and prove its correctness. More precisely, our protocol ensures message delivery in a connected network whenever the ratio of the maximum transmission range to the minimum transmission range is at most $\sqrt{2}$. This restriction guarantees that for any two hosts u and v, if there is another host w in the disk around the segment (u, v), then at least one of u and v will be aware of the existence of w. We note that the existence of such a protocol if R/r is unbounded is unlikely since allowing arbitrary values of R and r forces the consideration of graphs in which nodes are not connected to other nodes at a short distance from them while being connected to nodes further away. Like the protocols in [4, 8], our protocol may fail in a totally asynchronous environment, for example, if a mobile host can be arbitrarily slow while executing the protocol, and makes its neighbors aware of its existence very late in the running of the protocol. On the other hand our protocol does not require full synchronization between the mobile hosts, and a real-time environment in which every atomic operation has a bounded execution time is enough to ensure its validity.

The paper is organized as follows. Section 2 defines the model of ad hoc networks we use. Section 3 presents our protocol, while Section 4 proves its correctness. Discussion of the performance of the resulting routing protocol is in Section 5. Finally Section 6 contains some concluding remarks.

2. MODEL

We model an ad hoc network as follows. A set of mobile hosts is spread out in an unknown environment, modeled by the Euclidean plane. Two mobile hosts cannot communicate directly (that is, exchange messages) if their Euclidean distance exceeds a specified value called the maximum *range*. Conversely, two mobile hosts are always mutually reachable if their Euclidean distance is below a specified value, called the minimum range. These two thresholds depend on both the environment and the mobile hosts' technology. The ratio of the maximum range to the minimum range influences the efficiency of our protocol, but it is designed to work for all possible ratios smaller than $\sqrt{2}$.

We will use the following hypotheses and notation:

- Any mobile host knows the coordinates (x, y) of its position.
- The minimum transmission range is r, *i.e.* any two mobile hosts at distance $d \leq r$ are able to communicate directly.
- The maximum transmission range is R, *i.e.* no pair of mobile hosts at distance $d > R$ can communicate directly.
- Two mobile hosts at distance d, with $r < d \leq R$, may or may not be able to communicate directly.
- Communication links are *bidirectional*, i.e., if a mobile host u is able to receive signals from a mobile host v, then v is also able to receive signals from u.

The network is then represented by a geometric undirected graph, $G = (V, E)$, with vertices representing mobile hosts, and edges representing communication links. The set of vertices V is thus a set of points in the Euclidean plane. Let $d(u, v)$ be the Euclidean distance between the points u and v in the plane. The set of edges E satisfies $\{\{u, v\} : u, v \in V,\, d(u, v) \leq r\} \subseteq E$ and $\subseteq \{\{u, v\} : u, v \in V,\, d(u, v) \leq R\}$, that is, E contains all the pairs of mobile hosts at distance at most r, and some subset of the pairs at distance between r and R.

We note that the transmission conditions do not vary rapidly with time, compared to the speed of electronic communications. This implies that the network may change, for example, due to a change in the weather conditions, but it is at a time scale that allows an easy resetting of the network. Thus, we assume from now on that the connections between mobile hosts are fixed. However, the structure of these connections is not known, and it is the role of our protocol to ensure message delivery in this unknown network.

Initially, every mobile host is turned on, and becomes aware of its position in the plane (for example, by the use of GPS). The only knowledge initially given to a mobile host is its x- and y-coordinates in the plane.

3. A 3-PHASE ROUTING SCHEME

Our routing protocol is a routing scheme consisting of three phases. These phases are called the *completion* phase, the *extraction* phase, and the *routing* phase. The aim of the completion phase is to add edges, called *virtual edges* to the physical graph G, and hence to return a super-graph $S(G)$ of G. Without this phase, the planar graph resulting from the extraction phase would not necessarily be connected. Once the completion phase is done, the extraction phase is executed, and it returns a connected and planar spanning subgraph of $S(G)$. Finally, the routing phase performs message delivery between mobile hosts.

The completion phase is a new contribution, and is described in detail. The extraction phase is roughly the same as the Gabriel Graph construction of [4], and therefore its description is only sketched. Only the modifications induced by the virtual edges are described in detail as they are original. The routing phase is basically the same as the routing phase in [4, 8], and alternates greedy routing with perimeter routing, i.e., routing around the faces of a planar graph using the right-hand-rule. Thus we include only a brief discussion of the routing phase.

All computations in all phases are local and do not require any central controller.

3.1 Completion phase

Notation. The *ball* with center c and radius ρ, denoted $B(c, \rho)$, is the set of points in the plane at distance at most ρ from c. The *disk* determined by two points u and v in the plane, denote $D_{u,v}$, is the set of points in the circle having the segment u, v as diameter.

Initially, every mobile host u broadcasts a request for the coordinates of its neighbors. Every mobile host v receiving this request immediately responds to u with its own coordinates. The nodes corresponding to the mobile hosts, and the edges between a mobile host and its neighbors form the graph G. Mobile host u then sets up a list $\mathcal{L}(u)$ containing all its neighbors in G. It marks any node added to $\mathcal{L}(u)$ as unprocessed. For every unprocessed $v \in \mathcal{L}(u)$, u starts "processing" the edge $\{u, v\}$ as follows.

Processing of $\{u, v\}$ by u. For every $w \in \mathcal{L}(u)$, if $w \in D_{u,v}$ and $w \notin B(v, r)$ then[1]

- u sends the message "(new,w)" to v, including the coordinates of w;
- u sends the message "(new,v)" to w, including the coordinates of v;
- mark v processed.

End-Processing.

In parallel with the processing of its incident edges, every node is ready to receive new nodes' coordinates from its neighbors, and to update its adjacency list $\mathcal{L}$.

Updating the adjacency list of u. Upon receipt of a message "(new,w)" from a neighbor v, u checks whether $w \in \mathcal{L}(u)$. If not, then u adds w to $\mathcal{L}(u)$ as an unprocessed vertex. The edge $\{u, w\}$ is then set up as a *virtual edge*, and w is a *virtual neighbor* of u. Node u stores the path $\mathbf{ve}(u, w) = (u, v, w)$ corresponding to the virtual edge $\{u, w\}$.

[1]This condition is easy to check since, if $u = (u_1, u_2)$ and $v = (v_1, v_2)$, then $D_{u,v}$ is the set of points (x, y) satisfying $x^2 - x(u_1+v_1) + u_1v_1 + y^2 - y(u_2+v_2) + u_2v_2 \leq 0$, and $B(v, r)$ is the set of points (x, y) satisfying $(x - v_1)^2 + (y - v_2)^2 \leq r^2$.

End-Updating.

Clearly, this process involves sending messages to virtual neighbors, that is to destinations which are not directly connected to the source. This requires a simple routing protocol called *virtual routing* to distinguish it from the routing protocol resulting from our 3-phase scheme. We describe hereafter the sending and the forwarding of messages in the virtual routing protocol.

Sending a message. If node u has to send a message M to a (possibly virtual) neighbor v, then u calls $\texttt{send}(v, M)$ which proceeds as follows:

- If v is a neighbor of u, then the message (v, M) is transmitted to v;
- Otherwise, v is a virtual neighbor of u, and, according to $\mathbf{ve}(u, v) = (u, w, v)$, u performs $\texttt{send}(w, (v, M))$.

End-Sending.

Note that w may in turn be a virtual neighbor of u, then u will repeat the process, and actually sends M to w' where $\mathbf{ve}(u, w) = (u, w', w)$. Eventually, the message will be sent through a "real" edge of G incident on u.

Forwarding a message. Upon receipt of a message (u, M) from a neighbor, node u proceeds as follows:

- If $M = (v, M')$, $v \neq u$, then u performs $\texttt{send}(v, M')$;
- Otherwise u stores M;

End-Forwarding.

When a node u has completed the processing of all its edges (virtual or not), that is, when all nodes in $\mathcal{L}(u)$ are marked processed, it enters the extraction phase. Note that u may keep on switching between the extraction phase and the completion phase in the sense that u may learn about the existence of a virtual neighbor v after it has already entered the extraction phase. In that case, u stops the extraction phase and comes back to the completion phase to process the edge $\{u, v\}$. We prove later that this does not affect the correctness of our protocol.

The graph obtained by adding virtual edges to G is denoted by $S(G)$. At the end of the completion phase, every node knows the coordinates of its neighbors in $S(G)$, and knows a path in G corresponding to each of its incident virtual edges. Note that $S(G)$ formally exists only after all nodes have terminated the completion phase, i.e., after all nodes learn about new (virtual) neighbors and whose corresponding (virtual) edges have been processed.

One could argue that the knowledge of the coordinates of nodes within radius R of a given node u could be sufficient and could be somehow obtained using GPS alone. However the knowledge of the existence of v in the neighborhood of u, without knowledge of a path between u and v is not sufficient for our purposes. Creating virtual edges, in the manner described above, enables the construction of these paths.

3.2 Extraction phase

The aim of this phase is to extract a connected planar spanning subgraph of $S(G)$. This is performed by extracting the *Gabriel graph* of $S(G)$, denoted by $\mathbf{GG}(S(G))$. The vertex set of the Gabriel graph is V, i.e., the set of nodes. An edge $\{u, v\}$ of $S(G)$, possibly virtual, belongs to $\mathbf{GG}(S(G))$ if and only if there is no other node $w \in D_{u,v}$. As stated earlier, if H is a unit graph then $\mathbf{GG}(H)$ is a connected planar spanning subgraph of H. We will show later that $\mathbf{GG}(S(G))$ is a connected planar spanning subgraph of $S(G)$.

More precisely, in the extraction phase every node u proceeds as follows. Let $\mathcal{L}'(u)$ be initially equal to $\mathcal{L}(u)$. For every $v \in \mathcal{L}'(u)$, u determines whether $\{u, v\}$ belongs to $\mathbf{GG}(S(G))$ as follows.

Validating $\{u, v\}$ by u. Node v is removed from the list $\mathcal{L}'(u)$. If there exists $w \in \mathcal{L}(u) \cap D_{u,v}$, then u "deletes" the edge $\{u, v\}$ in the sense that v is not a neighbor of u (virtual or otherwise) in $\mathbf{GG}(S(G))$. Otherwise the edge $\{u, v\}$ is kept in $\mathbf{GG}(S(G))$.
End-Validating.

Once $\mathcal{L}'(u)$ is empty, u has completed the extraction phase, and is ready to route messages in the network (that is, not only those corresponding to the setting up of the system during the completion phase. It then enters the routing phase, which is in fact the "steady state" of the protocol. However, if at any time u becomes aware of a new neighbor, it comes back to the completion phase, and from there to the extraction phase. Again, we will show that this does not affect the correctness of the protocol.

3.3 Routing phase

Once the completion and extraction phases are completed and $\mathbf{GG}(S(G))$ is set up, routing in G is performed according to the strategy described in [4], that is a combination of greedy and perimeter routing in $\mathbf{GG}(S(G))$ is applied. Of course, perimeter routing in $\mathbf{GG}(S(G))$ alone can be used as well. Greedy routing generally performs faster than the perimeter routing, but it may fail at some nodes whereas perimeter routing ensures message delivery. It is important to note that perimeter routing requires routing through virtual edges. This routing is performed using the virtual routing protocol described in the completion phase.

As pointed out before, the lack of synchronization between the nodes may cause a node u to enter the routing phase, and to learn later about the existence of a (virtual) neighbor v. Then u comes back to the completion phase to process the edge $\{u, v\}$. Once u completes the completion phase again, it will re-enter the extraction phase, and eventually the routing phase again. The discovery of new neighbors may happen several times, and u may go back and forth between the several phases until it eventually stabilizes in the routing phase (no new neighbor remains to be discovered).

If node u comes back to the completion phase, it carries on routing according to its possibly wrong routing table. As we will see, this may cause temporary livelock, but routing will eventually succeed. In fact, as will be discussed in Section 5, frequent alternations between the different phases of our protocol do not occur often, and can be caused only by some specific placements of mobile hosts that are unlikely to occur in practice.

4. PROOF OF CORRECTNESS

The correctness of the routing protocol alternating greedy routing with perimeter routing has been shown in [4, 8] whenever the underlying graph used for the perimeter routing is planar and connected. Therefore, in this section, we prove the planarity and the connectivity of the graph $\mathbf{GG}(S(G))$, and the termination of our 3-phase scheme. First, we need to show that the construction of $S(G)$ terminates in a valid state.

LEMMA 1. *If the vertex u learns about a virtual neighbor v, then v eventually learns about the existence of u. Furthermore, the virtual routing protocol ensures the correct delivery of messages through virtual edges.*

Remark. Note that we assumed bidirectional communication links as a part of our model specification. Thus if u knows its neighbor v in the original network G, then v knows u as well.

PROOF. A node u learns about the existence of a new virtual neighbor v during the processing of an edge e by a node different from u and v. There are two cases corresponding to the two messages in the procedure `processing`: either $e = \{u, w\}$, $v \in D_{u,w}$, w processing e, or $e = \{v, w\}$, $u \in D_{v,w}$, w processing e. In both cases, w will send to u the coordinates of v and to v the coordinates of u. It remains to check that these coordinates are eventually received by u and v, that is, that the virtual routing protocol ensures delivery of these coordinates.

By construction, every virtual edge of $S(G)$ has two paths in the network G associated with it, one path for every endpoint. Indeed, the paths associated with an edge $\{u, v\}$ are $P_{u,v} = (u, v)$, and $P_{v,u} = (v, u)$. Moreover, the paths associated with a virtual edge $\{u, v\}$ are

- $P_{u,v} = (P_{u,w}|P_{w,v})$ where w is the node from which u learnt about v, and "|" is the concatenation of two paths; and
- $P_{v,u} = (P_{v,w'}|P_{w',u})$ where w' is the node from which v learnt about u.

Note that $P_{u,v}$ and $P_{v,u}$ may not be symmetric as the actual composition of a path depends on who is the first neighbor to inform u and v respectively.

Next, we show that the virtual routing protocol routes a message M from u to v along $P_{u,v}$. The proof is by induction on the length ℓ of the path $P_{u,v}$. If $\ell = 1$ then $\{u, v\}$ is an edge in the original graph G, and the procedure `send` transmits the message M from u to v along $\{u, v\}$. Assume now $\ell > 1$. The edge $\{u, v\}$ is a virtual edge. The node u has stored $\mathtt{ve}(u, v) = (u, w, v)$, and calls $\mathtt{send}(w, (v, M))$. By the induction hypothesis, the message (v, M) will eventually reach node w. There the procedure `forwarding` is executed, that is, w will perform $\mathtt{send}(v, M)$. Again, by the induction hypothesis, the message (v, M) will eventually reach the node v where it will be stored. This completes the proof of correctness of the virtual routing protocol, and therefore of the lemma. □

Lemma 1 has the following simple consequence.

LEMMA 2. *The completion phase terminates, i.e., all nodes eventually complete the completion phase, and never come back to it.*

PROOF. Every node u processes at most k_u edges where k_u is the number of nodes contained in $B(u, R)$. The processing of an edge consists of sending a finite number of messages to a finite number of nodes, and will therefore terminate. Lemma 1 showed that the routing of these messages will also terminate. Finally once an edge has been processed, it does not need to be processed again. □

Lemma 1 shows that if a node u has to make a decision about whether or not the (possibly virtual) edge $\{u, v\}$ must be removed from $S(G)$, then v eventually has to make a decision as well. The next lemma shows that u and v will make the same decision.

LEMMA 3. *If u decides to remove the (possibly virtual) edge $\{u, v\}$ during the extraction phase, then v eventually removes $\{u, v\}$ as well.*

PROOF. By definition of the Gabriel graph construction, if u decides to remove the edge $\{u, v\}$, then there is a neighbor w of u in $D_{u,v}$. Now, since u is currently in the extraction phase, the edge $\{u, v\}$ has already been processed during the completion phase. During that processing, u sent the coordinates of w to v. From Lemma 1, these coordinates eventually reach v, and hence v eventually decides to remove $\{u, v\}$ from $S(G)$. □

It is important to note that it might be possible that v previously decided to keep $\{u, v\}$ in $S(G)$ because v was not yet aware of the existence of w when it considered $\{u, v\}$. However, our protocol ensures that as soon as v becomes aware of w (and this will eventually occur) it has to stop the extraction phase (even possibly the routing phase) and return to the completion phase to process $\{v, w\}$. Later, v has to reconsider whether or not the edge $\{u, v\}$ should be removed from $\mathbf{GG}(S(G))$. Once w is known, v eventually decides to remove that edge from $\mathbf{GG}(S(G))$.

LEMMA 4. *The extraction phase terminates, i.e., all nodes eventually complete the extraction phase, and never come back to it.*

PROOF. The extraction phase consists of taking a local decision about which edges should be deleted. This local decision process always terminates. We may enter the extraction phase several times because of the discovery of new neighbors. However, from Lemma 2, there is a time at which every node finishes the completion phase and will not come back to it any more. The extraction phase is then performed for the last time, and is not entered any more. □

LEMMA 5. *If G is connected and $R/r \leq \sqrt{2}$ then the graph obtained after the termination of the extraction phase is planar and connected. In other words, $\mathbf{GG}(S(G))$ is planar and connected.*

PROOF. Assume that $\mathbf{GG}(S(G))$ is not planar, and let $\{u,v\}$ and $\{u',v'\}$ be two crossing edges in $\mathbf{GG}(S(G))$. Consider the parallelogram formed by the nodes u, u', v, v'. Clearly, at least one of the angles in the parallelogram is at least $\pi/2$. We may assume w.l.o.g. that the angle $u, u', v \geq \pi/2$. Then u' is in $D_{u,v}$. If $d(u,u') \leq r$ or $d(v,u') \leq r$ then u' would be a neighbor of at least one of the two nodes u and v, and therefore, from Lemma 3, the edge $\{u,v\}$ would have been removed from $\mathbf{GG}(S(G))$. Therefore $d(u,u') > r$ and $d(v,u') > r$. Therefore, since $u' \in D_{u,v}$, we get $d(u,v) > \sqrt{2} \cdot r \geq R$. This contradicts the fact that any edge of G or any virtual edge constructed in the completion phase is of length at most R. Hence $\mathbf{GG}(S(G))$ is planar.

$S(G)$ is connected since G is connected and $S(G)$ is a supergraph of G. By Lemma 3, two neighboring nodes u and v make the same decision on whether the edge $\{u,v\}$ of $S(G)$ is kept in $\mathbf{GG}(S(G))$. Therefore, we can focus on the local connectivity of a node with its neighbors $S(G)$. An edge $\{u,v\}$ is removed from $S(G)$ by u if there is a node w in $D_{u,v}$ who u is aware of, that is, $\{u,w\} \in E(S(G))$. Node u informed v of w's coordinates when it processed $\{u,v\}$. Therefore v is also aware of w and (u,w,v) is an alternative path from u to v. Now, two problems may occur:

(1) It may happen that w removes the edge $\{w,v\}$ from $\mathbf{GG}(S(G))$ because there is a node w' in $D_{w,v}$. However, since $w \in D_{u,v}$, $w' \neq u$, and hence the alternative path for (w,v) does not use the edge $\{u,v\}$.

(2) It may also happen that u removes the edge $\{u,w\}$ because of w'' in $D_{u,w}$. However, since $w \in D_{u,v}$, $w'' \neq v$, and hence the alternative path for (u,w) does not use the edge $\{u,v\}$.

Therefore, for every edge removed from $S(G)$, there is an alternative path in $\mathbf{GG}(S(G))$ joining its two endpoints. Thus, $\mathbf{GG}(S(G))$ is connected whenever $S(G)$ is connected. □

THEOREM 1. *If $R/r \leq \sqrt{2}$ then the 3-phase routing scheme produces a routing protocol that allows delivery of messages from any source to any destination in any connected wireless ad hoc network.*

PROOF. Lemma 4 shows that all nodes eventually enter the routing phase and remain in it. Moreover, Lemma 5 shows that when nodes are all in the routing phase, they agree on a graph $\mathbf{GG}(S(G))$ which is planar and connected. Our routing protocol applies perimeter routing to $\mathbf{GG}(S(G))$ when greedy routing fails. It is known that perimeter routing ensures delivery of messages in a planar and connected graph. By Lemma 1, message delivery is ensured along each edge of $\mathbf{GG}(S(G))$. Therefore, once the routing phase is entered by all nodes, our whole routing protocol ensures delivery of messages between any pair of nodes in the network. □

5. DISCUSSION

This section discusses the performance of our 3-phase routing scheme. Our first result concerns the problem of routing along virtual edges. It shows that the route corresponding to a virtual edge cannot be arbitrarily large.

PROPERTY 1. *[Good news] Let λ be the minimum Euclidean distance between any two mobile hosts. If $R/r \leq \sqrt{2}$, then the length of the route in G corresponding to a virtual edge in $S(G)$ is at most $1 + \frac{R^2 - r^2}{\lambda^2}$.*

PROOF. First we show that if $\{u,v\}$ is a virtual edge in $S(G)$, then the length of the route in G between u and v is at most $1 + (R^2 - d^2(u,v))/\lambda^2$. Property 1 directly follows since if $\{u,v\}$ is a virtual edge, then $d(u,v) > r$. The proof is by induction on the length $\ell(e)$ of the route.

Let $e = \{u,v\}$ with $\ell(e) = 2$. We have $e = (u,w,v)$, for some w, with either $u \in D_{v,w}$ or $v \in D_{u,w}$. Assume $u \in D_{v,w}$. Since $d(u,w) \geq \lambda$, we get that $d^2(u,v) \leq d^2(v,w) - \lambda^2$. Therefore, $(R^2 - d^2(u,v))/\lambda^2 \geq 1$ as claimed. The case $v \in D_{u,w}$ is similar.

Let $e = \{u,v\}$ with $\ell(e) = k > 2$. We have $e = (u,w,v)$, for some w, with either $u \in D_{v,w}$ or $v \in D_{u,w}$. Moreover, since $R/r \leq \sqrt{2}$, if $u \in D_{v,w}$ then $\{u,w\}$ is an edge of G and $\ell(\{v,w\}) = k-1$, otherwise (i.e., $v \in D_{u,w}$), $\{v,w\}$ is an edge of G and $\ell(\{u,w\}) = k-1$. Assume the former. By the induction hypothesis, we have $k - 1 \leq 1 + (R^2 - d^2(v,w))/\lambda^2$. Moreover, $d^2(v,w) \geq d^2(u,v) + \lambda^2$. Therefore $k \leq 1 + (R^2 - d^2(u,v))/\lambda^2$ as claimed. The case $v \in D_{u,w}$ is similar. □

The above property implies that the values of R, r and λ can be used to find the length of the completion phase.

It is known that unit disk graphs with a restriction on the minimum distance between two nodes have nice algorithmic properties [7]. This seems to be also the case as far as efficient communications in wireless ad hoc networks are concerned. Indeed, the following is a direct corollary of Property 1.

COROLLARY 1. *If $R/r = 1 + \epsilon$, $0 < \epsilon \leq \sqrt{2} - 1$, then placing the mobile hosts so that no two mobile hosts are at distance less than $\sqrt{\epsilon} \cdot r$ ensures that the length of the route corresponding to a virtual edge is at most 4.*

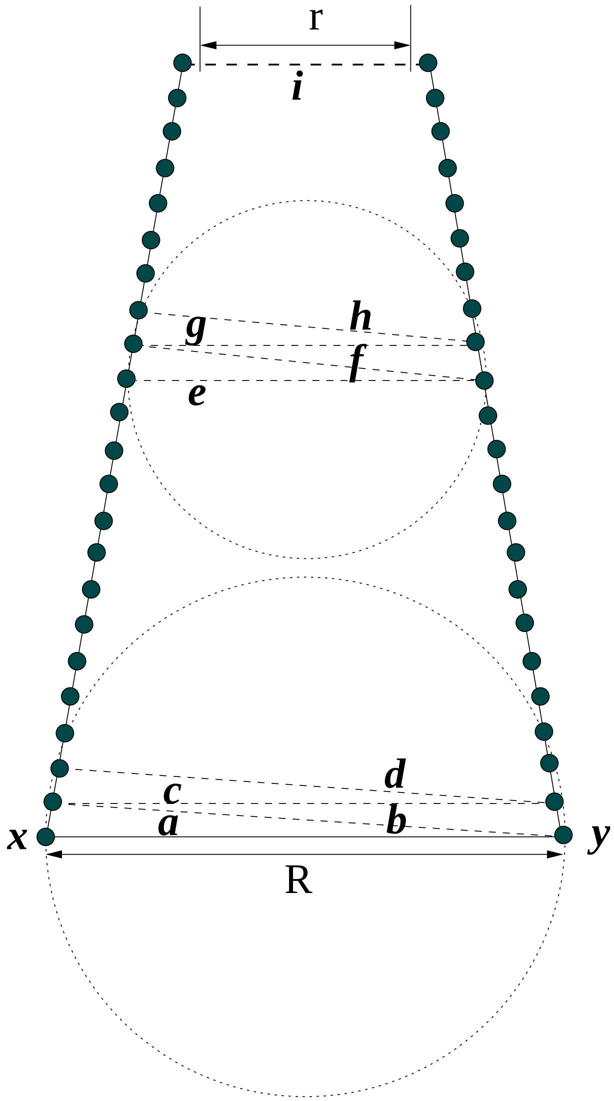

Figure 7: A configuration of the mobile hosts yielding "long" virtual edges.

For instance, if the variation in transmission ranges is at most 10%, then placing the mobile hosts at mutual distance at least $r/3$ ensures that the length of the route corresponding to a virtual edge is at most 4. However, there are cases for which our 3-phase scheme generates a high volume of communication, and for which there will be no benefit compared to flooding. Such a bad case can be summarized as follows:

PROPERTY 2. *[Bad news] For any $k > 0$, and any ratio $R/r \leq \sqrt{2}$, there is a configuration of the mobile hosts for which the path corresponding to a virtual edge is $\geq k$.*

PROOF. The "bad" configuration is depicted on Figure 7. There are two set of nodes, half of them on the left line and half of them on the right line. Each of the nodes on the left line is at distance larger than r from any of the nodes on the right line, and they do not see each other, except for the nodes x and y which are connected by the real edge a of G. The processing of a by x yields the virtual edge b. The processing of b by y yields the virtual edge c. The processing of the edge e yields a virtual edge f which yields a virtual edge g, and so on. The process ends with the construction of the virtual edge labeled i in Figure 7. Since the number of nodes on each side can be arbitrarily large, the route corresponding to the virtual edge i can be made arbitrarily large. □

Note that the importance of the bad news should be moderated in view of the good news in the sense that the good news requires mobile hosts to be not too close, whereas the bad example requires mobile hosts to be arbitrarily close. The reality is somewhat in between these two extreme cases. The bad case depicted in Figure 7 could correspond to a case where a large obstacle separates the nodes into two groups with only one real edge between the two groups. Clearly, the shortest communication paths between nodes of the two groups are long and the use of the virtual edge i only doubles the length of some of the communication paths.

Notice that the efficiency of the construction of $S(G)$ could be improved by applying heuristics. For example, edges can be processed in order of decreasing lengths.

6. CONCLUSIONS

We have proposed a robust position-based routing scheme allowing routing in ad hoc wireless networks in which transmission ranges may vary up to a ratio of $\sqrt{2}$. The value of $\sqrt{2}$ is the largest value that guarantees that for any two hosts u and v, if there is another host w in the disk around the segment (u, v), then at least one of u and v will be aware of the existence of w. Our scheme does not require flooding the network to set up the routing protocol, but rather every node performs a limited exploration of its locality. The degree of the locality of these explorations depends on the ratio of the maximal range to the minimum range, and on the minimum distance between two mobile hosts. It is expected to be small in practice. We are currently investigating the stretch factor of the routes used in the routing phase.

Many problems remain open, in particular:

1. What can we say if the ratio $R/r > \sqrt{2}$? (In this case, our scheme may fail because one cannot guarantee the planarity of $\mathbf{GG}(S(G))$.)

2. What can we say if the connections are not bidirectional, that is, u can receive signals from v but v cannot receive signals from u? (In this case, our scheme fails because the Gabriel graph construction requires symmetry between the two endpoints of an edge.)

3. Would it be possible to derive a routing scheme which ensures the termination of the set-up of the routing before the routing starts? (Our 3-phase scheme allows nodes to start routing messages while the set-up of the routing tables is not completed.)

7. ACKNOWLEDGMENTS

This research was done while the first two authors were visiting Concordia University in Montréal, the first author with support from NATO, and the second author with support from the RNRT program ROM. The research of the last two authors is supported by NSERC grants.

8. ADDITIONAL AUTHORS

Additional author: Jaroslav Opatrny
(Department of Computer Science, Concordia University, Montreal, Canada, email: `opatrny@cs.concordia.ca`)

9. REFERENCES

[1] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. A distance routing effect for mobility (DREAM). In 4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98), pages 76-84, 1998.

[2] P. Bose, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, L. Jacobsen, A. López-Ortiz, P. Morin, and J. Munro. Online routing in convex subdivisions. In 11th Annual International Symposium on Algorithms and Computation (ISAAC '00), 2000.

[3] P. Bose and P. Morin. Online routing in triangulations. In 10th Annual International Symposium on Algorithms and Computation (ISAAC '99), pages 113–122, 1999.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In 3rd Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM '99), pages 48-55, 1999. (Journal version to appear in *Telecommunication Systems.*)

[5] J. Broch, D. Johnson, and D. Maltz. The dynamic source routing protocol for mobile ad hoc networks; Internet-draft, draft-ietf-manet-dsr-00.txt, March 1998.

[6] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. Systematic Zoology 18, pages 259-278, 1969.

[7] H. Hunt III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, February 1998.

[8] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In 6th ACM Conference on Mobile Computing and Networking (Mobicom '00), 2000.

[9] Y.-B. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In 4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98), pages 66-75, 1998.

[10] Y.-B. Ko and N. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), 1999.

[11] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In Canadian Conference on Computational Geometry (CCCG '99), pages 51–54, 1999.

[12] X. Lin and I. Stojmenović GEDIR: Loop-free location based routing in wireless networks. In IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS '99), pages 1025-1028, 1999.

[13] A. Okabe, B. Boots, and S. Sugihara. Spatial tesselations: concepts and applications of Voronoi diagrams. John Wiley and Sons, 1992.

[14] V. Park and S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks; in Proceedings of INFOCOM'97, pp. 1405-1413, 1997.

[15] C. Perkins. Ad hoc on demand distance vector (AODV) routing; Internet draft, draft-ietf-manet-aodv-00.txt, November 1997.

[16] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In Proceedings of SIGCOMM 94, pages 234-244, August 1994.