Guaranteed-delivery Geographic Routing under Uncertain Node Locations

Stefan Funke
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
Email: funke@mpi-inf.mpg.de

Nikola Milosavljević
Computer Science Department
Stanford University
Stanford, CA, U.S.A.
EMail: nikolam@cs.stanford.edu

***Abstract*—Geographic routing protocols like GOAFR or GPSR rely on *exact* location information at the nodes, because when the greedy routing phase gets stuck at a local minimum, they require, as a fallback, a planar subgraph whose identification, in all existing methods, depends on exact node positions. In practice, however, location information at the network nodes is hardly precise; be it because the employed location hardware, such as GPS, exhibits an inherent measurement imprecision, or because the localization protocols which estimate positions of the network nodes cannot do so without errors.**

In this paper we propose a novel naming and routing scheme that can handle the uncertainty in location information. It is based on a macroscopic variant of geographic greedy routing, as well as a macroscopic planarization of the communication graph. If an upper bound on the deviation from true node locations is available, our routing protocol guarantees delivery of messages. Due to its macroscopic view, our routing scheme also produces shorter and more load-balanced paths than common geographic routing schemes, in particular in sparsely connected networks or in the presence of obstacles.

I. Introduction

For many applications of wireless sensor networks it is essential that the wireless nodes be aware of their geographic location. A notable example is geographic routing, which has been extensively studied in the past [1], [2], [3], [4], [5], [6], and proved to be very useful in practice because of its simplicity, scalability and low routing overhead. Geographic routing is also a basis for other higher-level applications of sensor networks, such as the data-centric storage, or tracking and surveillance.

However, the assumption that nodes have their *exact* geographic location information available is often unwarranted. Even with onboard GPS receivers, there is a certain measurement error which causes the reported geographic location to be only near the true location. Also very often – in particular in wireless sensor network applications where thousands of nodes are deployed – one cannot afford equipping every sensor node with a positioning device due to high cost. Furthermore, such a device would waste energy, or even fail to work in the areas of bad reception or indoors. Another way in which nodes can discover their locations is using one of the localization algorithms to infer the location from a small set of anchor nodes, equipped with GPS receivers. This solution is even less satisfactory, since the performance of localization algorithms heavily depends on the distribution and number of anchors, as well as the connectivity of the network as a whole.

For geographic routing protocols, imprecise location information can have devastating consequences. In particular, when the network is sparse or in the presence of obstacles, the greedy geographic routing phase frequently encounters local minima, where a packet gets stuck due to all neighboring nodes being further away from the target than its current location. In this case, a recovery phase has to be employed, guiding the packet out of the local minimum and towards the target node. This recovery phase is based on a planarization of the communication graph, which essentially allows the packet to be routed around communication voids in the network. In the presence of imprecise location information, typical planar graph constructions like the Gabriel graph[1] do not necessarily produce a connected planar subgraph of the communication graph. See for example Figure 1. Here we have drawn three nodes in the plane, all with a communication radius of 1. Assume that the locations of nodes P and Q are known precisely. The position of S is incorrectly estimated to be at S', though. Note that the absolute deviation from the true position is less than $1/3$. The communication graph of these three nodes contains only two edges, (P, Q) and (Q, S). Unfortunately, due to the imprecise location information of node S, the edge (P, Q) does not pass the Gabriel edge condition (having an empty diametral ball) and hence is deleted. The remaining graph is not connected anymore and cannot be used as a reliable fallback in case when the geographic greedy routing phase gets stuck. Furthermore, once a message gets very close to its target, imprecise node locations are of no use in delivering the message, hence some different strategy, such as local flooding, has to be employed.

On the other hand, for some sensor network applications it is not necessary to know the *true* locations of sensor nodes, but instead some other set of *invented* locations can serve the purpose; the made up coordinates may or may not have to resemble the original ones in one way or another, depending on the application. The true coordinates are typically less important in cases when they are only internally used by the

[1]Two points/nodes in the plane p,q are neighbors in the Gabriel graph if the ball whose equator is $\overline{pq}$ is empty of other points/nodes.

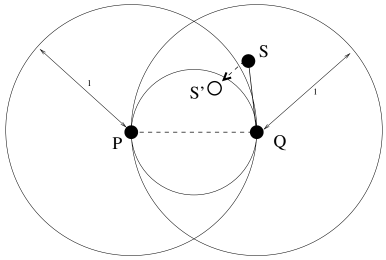


Fig. 1. Non-connectedness of the Gabriel graph construction in case of imprecise node locations.

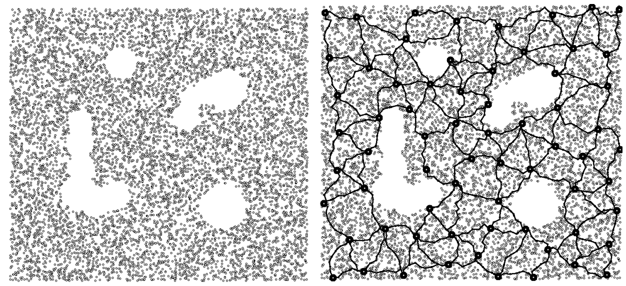


Fig. 2. A complex network topology (left), and the planar *Combinatorial Delaunay Map (CDM)* computed by our algorithm (right).

algorithm and not communicated to the user; for example in the case of geographic hashing for data-centric storage. On the other hand, they are clearly indispensable in tracking applications, where in fact this is the main information that a user wants to know about.

Recently a number of methods for computing these so called *virtual coordinates* has been proposed in the literature. These methods typically use only connectivity information, which is in in most cases readily available in a wireless network, see for example [7], [8], [9]. The problem with all these approaches is that the computation of these virtual coordinates is computationally rather challenging and in all the approaches requires at least in some steps a global view on the network when computing a planar graph embedding. The approach that we present in this paper can be seen as a way to avoid the expensive and centralized embedding computation by using approximate location information that is often available at the nodes.

Another approach to avoid the complexity of global embedding is suitably partitioning the domain and constructing a set of local embeddings which are then connected using some other, perhaps non-geometric routing method. The *landmark based method* called GLIDER [10], proposes a two-tier scheme, targeted at making the routing infrastructure stable under small fluctuations of the network connectivity, which are inevitable in practice. It achieves this goal by having a preprocessing phase in which a sparse set of *landmark nodes* is selected to serve as a rough guideline for high-level global routing between distant parts of the network. The assumption is that the connectivity of such a sparse set, which captures the global topology of the network, tends to be more robust to local link variability. The routing is then performed by using the landmark graph and the associated routing table (which every node can afford to store locally during the preprocessing) to plan the global route, and realizing this route in the original real-world network (in a way that takes care not to overload the landmark nodes). The set of landmarks has to be sparse, in order to be of manageable size for a single sensor to store. On the other hand, The number of landmarks is also constrained by the complexity of the sensor field that the landmark graph has to capture (in particular, the number of holes and passages). Clearly, these two requirements are often in collision.

In this paper we propose a method that combines the ideas from several of the above approaches. It takes a high-level view on the topology of the sensor field, by adopting a landmark-based approach, and thus inheriting the robustness and load-balancing properties of e.g. GLIDER. On the other hand, we remove the sparseness restriction on the landmarks, that is we can afford to have as many of them as required by the amount of topological features of the network. We achieve this by not having to store the landmark graph and its routing information in each node; instead, we *use geometric routing to navigate the landmark graph*. What makes this possible is the fact that our landmark graph is guaranteed to be *planar*. We take advantage of the fact that even imprecise location information suffices to embed such a graph in a completely localized fashion.

II. Overview

Unless noted otherwise, we assume that the sensors are represented by points in the plane and can communicate with each other according to the unit disk graph [11] or quasi-unit-disk graph [12] model of connectivity. The *quasi-unit-disk* graph with parameter α for a set of points in the plane certainly has an edge between all nodes that have distance at most $1/\alpha$ and certainly no edge between pairs of nodes at distance more than 1. For nodes at distance between $1/\alpha$ and 1 the the presence of an edge is uncertain. In our exposition we furthermore only deal with the static case, i.e. stationary network nodes, as this is typical in wireless sensor network applications.

Our starting point is the GLIDER method of Fang *et al.* [10], in the sense that we also build a two-layer landmark-based structure. We extend and modify their approach in the part that deals with the upper layer of the hierarchy. The authors of [10] do not propose a specific landmark selection scheme, but they emphasize that the set of landmarks should be sparse, because its main purpose is to capture the large scale topological features of the sensor field, such as the large holes, which are assumed to be few. Furthermore, they use the fact that the global topology is likely to remain stable over extended periods of time, given the typical variations in network connectivity. Their routing algorithm has a proactive global planning phase that uses a global routing table stored at each node's memory. Clearly, sparsity and stability of the landmark graph are the key properties that enable such an approach.

In this work we take a slightly different approach in the landmark selection strategy and the construction of the landmark graph. First, we allow our landmark set to be much denser. In the GLIDER terminology, this corresponds to partitioning the sensor field into many very small "routable tiles", where each tile is a set of nodes that share the closest (in the graph distance) landmark. Smaller tiles means finer resolution, and the ability to capture smaller topological features. On the other hand, the complexity of the landmark complex becomes prohibitively high, so the routing table approach to global planning, employed by GLIDER, is not scalable any more.

We get around this problem by making sure that the landmark graph that we construct is planar. The graph constructed by GLIDER is a variation of the *combinatorial Delaunay graph* (CDG), which was proposed in [13] as a way to extend the classical notion of Delaunay triangulation from planar geometry to the unit-disk graph setting. The idea is to approximate the Voronoi and the Delaunay diagram of a sparse set of landmark nodes (analogous to a set of points in the plane), using the rest of the nodes as a sampling of the plane. Then the nodes are partitioned into Voronoi tiles, such that the nodes in a single tile share the closest landmark (measured in the graph distance). Variations of this construction have been studied in the past under the name *graph Voronoi diagram* [14], [15]. Finally, the CDG is a graph induced by the adjacencies of the Voronoi tiles associated with distinct landmarks.

Unfortunately, the CDG is not a triangulation, in fact it is not even planar in general. The typical reason for violating planarity is that many spurious Delaunay edges (tile adjacencies) are detected in the "degenerate" regions of the network, i.e. those that are roughly equally distant from more than three landmarks. While this type of degeneracy can often be considered an extremely rare, or easy to handle event in the continuous geometry, in the graph setting it is not the case.

We present a completely localized and distributed procedure for computing a landmark graph which is provably planar. We adopt a similar approach of trying to mimic planar geometry, but we take special care that the degeneracies as described above are not considered as valid. Then, we construct a planar embedding of the landmark graph. We argue that even imprecise location information at the landmark nodes suffices to compute such a planar embedding in a completely localized fashion. The resulting embedding faithfully resembles the true geometry and topology of the sensor field. Finally, we use the approximate coordinates of the landmark nodes as their virtual coordinates for global planning purposes, and use standard geographic routing algorithms to route packets at the macroscopic level, among the tiles. At the microscopic level – when forwarding a packet from a tile to an adjacent tile as determined by the geographic routing algorithm – we employ the same strategy as GLIDER by following gradient lines to landmarks of adjacent tiles.

Note that this completely eliminates the need for keeping any global state information stored at any of the nodes.

In Figure 2, left, we see a network of complex topology including holes. Using the structure of the connectivity graph and the approximate geographic locations of the nodes, our algorithm computes a *planar Combinatorial Delaunay Map (CDM)* (and its embedding), as can be seen on the right. The CDM is navigated using a macroscopic version of geographic routing with the fallback routine based on the planar embedding of the CDM.

The rest of the paper is organized as follows. In Section III we introduce a variant of greedy geographic routing that works on the macroscopic level of the landmark graph, at that point still assuming that exact geographic location information is available. In Section IV we describe a distributed construction of a provably planar substructure of the combinatorial Delaunay graph, which we call *Combinatorial Delaunay Map (CDM)*. Due to space restrictions we leave out the proofs, which can be found in [9]. We show how the combinatorial embedding of the CDM can be computed even using only approximate location information at the landmark nodes. Finally, in Section V we compare our new approach with known routing protocols like GLIDER and GPSR.

III. Macroscopic Geographic Greedy Routing

Let us for now assume that all the nodes in our wireless network by some means are aware of their true geographic location. In that case, very attractive routing protocols are based on geographic routing, as pioneered in GPSR, GFG and refined later e.g. in AFR [6], GOAFR [4] and GOAFR+ [5]. The idea of greedy geographic routing is very simple: when sending a packet to a destination, the node currently holding the packet simply forwards it to the node among its one-hop neighbors which is closest to the destination (based on the Euclidean distance given by the geographic locations). It can happen, though, that in presence of holes in the network a packet gets stuck in a local minimum. Different methods have been proposed to overcome this problem. The best known among those is probably GPSR, which exhibits a planar subgraph of the connectivity graph and uses perimeter forwarding when the greedy phase gets stuck. What makes GPSR and other geographic routing schemes so attractive is the fact that, in particular for dense network deployments, the produced paths are very close to optimum, still no auxiliary routing structures have to be maintained.

The quality of the paths produced by geographic routing protocols decreases, though, in the presence of holes or for low node densities (as this induces small holes everywhere in the network). In these situations GPSR for example often has to switch from the greedy phase to perimeter routing, which not only tends to produce longer paths than necessary, but also leads to load imbalance in particular on the hole boundary nodes. One can partly remedy this problem by considering not only a 1-hop neighborhood but a k-hop neighborhood when determining the node where to forward the packet. If the size of the holes is below the chosen neighborhood size k, the greedy phase will still succeed and not get stuck in a local minimum. The problem with this approach is that the required storage per node to keep track of the k-hop neighborhood increases drastically with k (it is about quadratic in k).

In this section we introduce a variant of geographic greedy routing – we call it *Macroscopic Geographic Greedy Routing (MGGR)* – that is also less susceptible to small holes, but does not require additional memory at the network nodes to store an extended neighborhood. The core of our approach is to first compute a partition of the network into small *tiles* by choosing a set of landmarks and assigning each node to its closest (in hop-distance) landmark. The same idea is employed by the GLIDER routing protocol, with the crucial difference, though, that for GLIDER one can only afford to create a *small* number of landmarks, since the resulting landmark Voronoi complex (LVC) or its respective dual, the combinatorial Delaunay graph (CDG), which captures the adjacencies of the tiles, has to be stored at *every* single network node. For MGGR the navigation between the tiles will not be based on the global availability of the CDG but on location information, so the maintenance and storage overhead does not grow with the number of tiles. Like in GLIDER, special treatment is necessary after the packet has reached the tile of the target node. For GLIDER, a local coordinate system is defined within the tile. The packet is then led to its final destination by the local coordinates. If this fails, the whole tile is flooded, which can be quite expensive, as the tiles are rather big for large networks. In case of MGGR, due to the increased number of landmarks, the tiles are relatively small, so flooding a tile or broadcasting a message across a tile via a backbone (e.g. a connected dominating set) is a relatively cheap operation.

A. Landmark Selection and Inter-tile Connectivity

Moving between the two extremes – each node is a landmark vs. picking only a constant number of landmarks independent of the network size – interpolates the resulting combinatorial structure between the original communication graph (as used for GPSR) and the combinatorial Delaunay graph (as used in GLIDER). We certainly aim to choose a large number of landmarks as we want to bound the effort for delivering a packet in its final tile. On the other hand we don't want to have the tiles too small as MGGR then becomes more susceptible to small local minima/holes, like the original GPSR. There is another argument for choosing not too many landmarks/too small tiles which will only become apparent in the next section where we discuss the case when only imprecise location information is available.

Our approach is to have the inter-landmark distance be constant, i.e. independent of the network size. To this end, we fix a small constant k, e.g. $k = 5$, and select the set of landmarks to be a k-hop independent set of nodes. For this we use a very simple greedy algorithm. Initially, all nodes are active. Active nodes decide to become landmarks (join the independent set) asynchronously. When a node becomes a landmark, it broadcasts an IN message within its k-hop neighborhood. All the recipients of the message become inactive, i.e. lose the ability to become landmarks. A deactivated node broadcasts an OUT message within its k-hop neighborhood. The process continues until all nodes are either landmarks or inactive.

To show that the output is indeed a k-hop independent set, it suffices to ensure that a node cannot receive a deactivation message after it has already joined the landmark set. This is achieved by requiring that the nodes join the independent set in the order of increasing IDs, i.e. a node is allowed to become a landmark only after it has received either an IN or an OUT message from all its k-hop neighbors with lower IDs.

We then compute the *graph Voronoi diagram* [14] of the communication graph with respect to the set of landmarks selected as above. The graph Voronoi diagram is a graph analogue of the corresponding object defined in geometry. Informally, the Voronoi diagram of a graph with respect to a subset of its vertices (the landmarks) is a partition of the vertex set into disjoint subsets (Voronoi regions or *tiles*) according to which landmark is closest to the vertices of a given subset. The distances are defined in a standard graph-theoretic sense, as shortest path lengths on the unweighted communication graph. Note that Voronoi regions and landmarks are in one to one correspondence, as in the geometric case.

Finally, we store for each node v in tile τ_v the distance to and coordinate of each landmark whose respective tile is adjacent[2] to τ_v. For communication graphs of bounded doubling dimension – as they typically arise in the context of wireless networks – there are only few adjacent tiles, hence there is only very little additional storage required at each node. The *combinatorial Delaunay graph (CDG)* is then the graph which has a node for each landmark and an edge between two landmarks if and only if the respective tiles are adjacent. See Figure 3 for a depiction of these concepts: on the left, the whole network with a set of landmarks selected as a maximal k-hop independent set; in the middle, the induced graph Voronoi diagram (in fact only edges that connect two nodes in two different tiles); and on the right, the dual structure, the combinatorial Delaunay graph (CDG), which exhibits non-planarities in several places.

The construction – apart from the fact that we create many tiles of constant diameter instead of a constant number of tiles of rather large diameter – is exactly the same as in GLIDER; hence we can employ their distributed construction, only skipping the global distribution of the CDG across the whole network. And likewise we use the distance information to landmarks of adjacent tiles to forward a packet from one tile t_1 to an adjacent tile t_2 by just following the gradient of the respective distance function towards the landmark of t_2 until crossing the boundary to t_2.

B. Naming and Greedy Routing

The naming scheme for the resulting network is rather straightforward: a node v with unique ID id is assigned the name (p, id) where p denotes the position of the landmark

[2] A tile τ is adjacent to a tile τ' if there exist nodes $v \in \tau$, $v' \in \tau'$ and (v, v') is an edge in the communication graph.

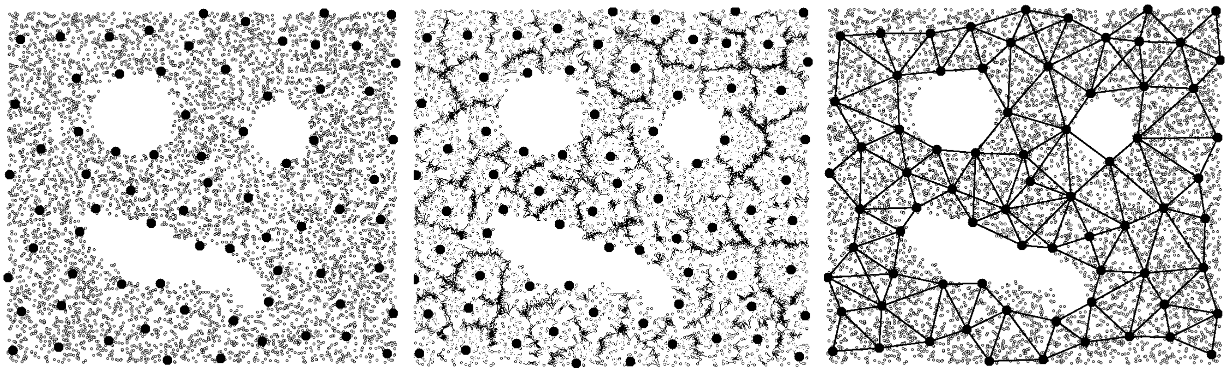


Fig. 3. A network topology with a set of landmarks, the induced graph Voronoi diagram, and its dual combinatorial Delaunay graph(CDG).

owning the tile that contains v. Clearly, the resulting names are unique if the node IDs were unique in the first place. Several nodes can have the same first component in their name — these are exactly all the nodes that belong to the same tile.

Greedy routing then follows the same scheme as GPSR but on a macroscopic level. Let us assume a packet needs to be sent from a node with name (p_s, id_s) to a node with name (p_t, id_t). Assume first that $p_s \neq p_t$, i.e. the target node resides in tile τ_t, the source node resides in tile τ_s with $\tau_t \neq \tau_s$. As long as the packet is at a node v in τ_s with respective first naming component p_s, the node inspects the coordinates $p_1, p_2, \ldots p_l$ of all landmarks of the adjacent tiles $\tau_1, \tau_2, \ldots \tau_l$ and selects some tile τ_i as target tile if the *Euclidean* distance from p_t to p_i is smaller than that to p_s. It can reach the boundary of τ_i by just following the distance gradient stored for the landmark of τ_i. Once it reaches τ_i it again inspects the neighboring tiles of τ_i whether for any of them the respective landmark is closer to p_t. The process is repeated until the packet hopefully reaches tile τ_t. There remain two issues to address. What happens if none of the landmarks of the adjacent tiles are geographically closer to the target landmark than the landmark of the current tile? And once the packet reaches the target tile τ_t, how is the packet delivered to the respective target node (p_t, id_t)?

C. Intra-Tile Connectivity

Let us first discuss the packet delivery in the final tile. By construction via a maximal k-hop independent set for small k (e.g. $k = 5$), our tiles are very small (their diameter is bounded by $2k$), that is we can either afford flooding the whole tile – which is much cheaper than in GLIDER, where the tile size essentially grows with the network size if the number of landmarks is kept constant – or we can use one of the many known algorithms to construct a connected dominating set (CDS)[3] within each tile and use this as a backbone to spread the packet to all nodes in that tile. Note that the CDS only has to cover the nodes within one small tile and hence can be computed completely locally. Hence when the packet enters the final tile in some node v, either v is adjacent to a node in the CDS backbone or already part of the CDS backbone. In the former case it relays the packet to one of its adjacent nodes contained in the CDS backbone (which has to exist as it is a connected *dominating* set). Then the packet is propagated through the CDS backbone where essentially every node $w \in$ CDS retransmits the packet once. By the domination property all nodes within the tile are guaranteed to receive the packet. Due to the sparsity of the CDS, interference can also be kept under control much easier than in case of flooding the tile in an uncontrolled fashion. A connected dominating set can be easily computed locally within each (small!) tile using known algorithms like [16].

[3]A connected dominating set for a graph $G(V, E)$ is a subset $D \subseteq V$ such that $\forall v \in V : v \in D$ or $w \in D$ for some w such that $(v, w) \in E$.

D. Planar Graph Construction

As in case of the "non-macroscopic" GPSR it may happen that a packet that is currently in some tile τ_i with landmark at position p_i realizes that all of the landmarks of adjacent tiles are further away from the target landmark than p_i. That is, the packet is stuck at a *local minimum*. To overcome this problem, GPSR switches to the perimeter routing phase where, with the help of a planar subgraph of the original unit-disk communication graph, the packet is routed around problematic local minima or holes. The important fact here is that perimeter routing requires the identification of a *planar* subgraph of the network in which the greedy routing takes place (the whole communication graph in case of GPSR). To be more precise, also GPSR requires a geometric *embedding* of this planar subgraph. Since in our case GPSR operates on the macroscopic level of tiles or respective landmarks, we essentially need to exhibit and construct a planar subgraph of the adjacency graph of all the landmarks. This can be accomplished by the following construction. First create an edge between two landmarks if their respective tiles are adjacent. The resulting graph is the above mentioned combinatorial Delaunay graph as used in the GLIDER approach. Unfortunately, this graph is typically not planar. However, since the actual edges in the CDG cannot be longer than $2k$, a simple inspection within a local neighborhood can identify intersecting pairs of edges, one of which can then be removed. This technique has been exploited in [17], but of course the search performed by their algorithm may not be local. So when during the greedy phase a packet gets stuck in a local minimum, this planar graph is used to recover. We will not elaborate on this planarization approach but instead present an alternative way to planarize the CDG which also works for *imprecise* node locations.

E. Summary

In this section we combined ideas from geographic routing protocols, like GPSR, with landmark based routing schemes, like GLIDER. The rationale behind that is to improve the behavior when local minima are encountered during the greedy routing phase; this still poses a problem for geographic routing protocols like GPSR, especially in scenarios where the network exhibits many holes or has a low-density communication graph. While we expect the greedy phase in this macroscopic version of geographic routing to encounter fewer local minima, we still have the fallback in the form of a planar substructure on the CDG, instrumented by perimeter routing to guarantee message delivery. In our experimental section we will see that MGGR leads to a much higher delivery rate of the greedy phase than GPSR without perimeter routing. Note that up to now we were assuming that exact location information is available at the selected landmark nodes.

IV. Planarization and Embedding of the Combinatorial Delaunay Graph

Let us now switch to the scenario where only *approximate* geographic location information is available. Most of the construction described in the previous section still applies. One essential ingredient is missing, though: how can we prune the combinatorial Delaunay graph to guarantee planarity based on approximate location information only?

In a first step we will sketch how a planar subgraph of the CDG can be extracted – we call it *combinatorial Delaunay map (CDM)*. Then we show how a combinatorial embedding[4] of the CDM can be computed using the approximate location information at the landmark nodes. This combinatorial embedding allows us to use the recovery/fallback protocols as described e.g. in GPSR or GOAFR.

A. Planar Graph Extraction

The idea for the construction and the main properties of our planar graph are largely derived from geometric intuition. To be specific, the planarity follows from the fact that our CDM is the *dual graph* of a suitably defined partition of the plane into simply connected disjoint regions. In the following, we define such a planar partition based on the landmark set, and propose a method for identifying a subset of edges of the combinatorial Delaunay graph using only the information available in the graph connectivity. The whole reasoning is based on the fact that the original communication graph is not an arbitrary graph but in some way resembles the geometry of the underlying domain by being either a unit-disk or quasi-unit-disk graph. First we introduce a *labeling of the communication graph* for a given set of landmarks.

Definition 1:

(i) Consider a landmark a and a vertex v. We say that v is an a-vertex if a is one of the landmarks which are closest to v, and it has the smallest ID among all such landmarks.

(ii) Consider arbitrary landmarks a, b and an edge $e = (u, v)$. We say that e is an a-edge if both u and v are a-vertices. We say that e is an ab-edge if u is an a-vertex and v is a b-vertex or vice versa.

Clearly, this rule assigns a unique label to each vertex and edge, due to the uniqueness of nodes' IDs. Also note that any landmark a is an a-vertex. Next we present a criterion for making two landmarks adjacent in the CDM.

Definition 2: Landmarks a and b are adjacent in the CDM if there exists a path from a to b whose 1-hop neighborhood (including the path itself) consists only of a and b vertices, and such that in the ordering of the nodes on the path (starting with a and ending with b) all a-nodes precede all b-nodes. We call such a path *witness path* for the adjacency between a and b. Note that the CDG is actually defined in a very similar way with the only difference that the 1-hop neighborhood of the path is not cared about.

Due to space restrictions we cannot elaborate on the proof of why this construction yields a planar subgraph of the CDM. Instead, we refer to [9], where the full proof is given in detail, and only cite the main result of [9]:

Theorem 1: The combinatorial Delaunay map (CDM) built using the rule of Definition 2 is planar for any quasi-unit disk graph with $\alpha \leq \sqrt{2}$.

B. Embedding the Combinatorial Delaunay Map

The goal of the embedding phase is to determine the clockwise order of the paths that witnessed the adjacencies of any vertex in the CDM, that is, to determine the combinatorial embedding of the CDM.

Consider some landmark a and its neighboring landmarks in the CDM, one of which is b. For an edge (a, b) in the CDM we define its *cone*, denoted by $\mathrm{cone}(a, b)$, to be the angle under which a sees all nodes on the witness path from a to b outside a's tile. For a landmark a we consider the set E_a of all adjacent edges in the CDM; we determine a maximal subset E'_a of those edges such that their respective cones are mutually disjoint. We say that landmark a *supports* the edge set E'_a. We keep an edge (a, b) from the CDM if and only if it is supported by both a and b. In the following, let us only consider such edges supported by both endpoints, and the associated *Refined Combinatorial Delaunay Map (RCDM)*. By construction, it is obvious that the order of the angles around each landmark gives us the circular order of the surviving adjacencies. That is, we have determined the combinatorial embedding of the RCDM. Observe that this process is completely localized, since a landmark has to inspect only adjacent landmarks which are at most $2k$ hops away, and k is chosen to be a small constant.

[4]A combinatorial embedding of a planar graph is given by its nodes and edges, as well as a cyclic ordering of the edges around each vertex in some planar embedding of the graph. Given a combinatorial embedding, the face cycles can be easily traversed.

So far we have still neglected the fact that only *approximate* node locations are available. Assume now that node locations are not given exactly, but with some uncertainty δ, that is, the true position of a landmark might be δ away from the location reported to us.

1) Choice of the inter-landmark distance based on location uncertainty: The idea of how to deal with uncertain node locations is rather trivial. We simply put a ball of radius δ around each node of the witness path. $\mathrm{cone}(a, b)$ is then defined as the angle under which a sees all the *balls* of radius δ around the nodes on the witness path from a to b outside a's tile. That, of course, widens $\mathrm{cone}(a, b)$, but not by much, provided that k – which determines the minimum distance between adjacent landmarks – is chosen large enough. So, the more uncertain the node locations are – i.e. the larger δ – the larger one has to pick k — in our experiments we obtained very good performance (that is, most edges of the CDM survived the angle test) by setting $k = 5 \max(1, \delta/2)$. Choosing k larger essentially makes the angles under which the witness path parts are seen smaller (since they are at least k hops away).

2) Dealing with disconnectedness of the CDM: Using our rules for pruning adjacencies from the CDM, it might happen in theory that a landmark a chosen by our algorithm turns out to be disconnected from the rest of the CDM. In such cases, we would delete a and assign its tile accordingly to the neighboring landmarks. In practice, though, for all the network deployments we have considered, this is not an issue.

C. Embedding in the Absence of Location Information

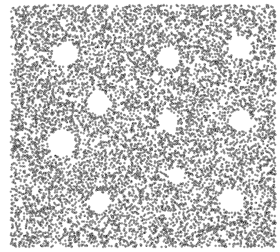
Our procedure for extracting a planar graph does not rely on any location information, only the embedding phase does. It is actually possible to derive an embedding of the extracted planar graph without using any location information, [9] is an example. Unfortunately, these methods are typically more involved and require non-local computation.

V. Simulation Results

To evaluate the performance of our routing scheme, we performed a set of computer-simulated experiments. We compared our routing scheme to GLIDER and GPSR in terms of delivery success rate, path length, message load on the nodes and the number of routing messages sent. In particular, we were interested in the comparison with GPSR in the case of uncertain node locations (note that imprecision of the node locations has no effect on GLIDER, since GLIDER does not use location information).

Our simulator (written in C++) is not packet-based, and thus it does not take into account some issues that occur in practice (i.e. medium access and message loss). However, we feel that these factors would have similar impact on all algorithms, and thus would not significantly affect the relative performance.

The localized Delaunay graph [18] is used for face routing in GPSR. In all the experiments involving GLIDER, we use a fixed number of 20 landmarks, which is consistent with GLIDER's philosophy of sparse landmark graphs.



	GPSR	MGGR
$\delta = 0$	0.613	0.929
$\delta = 0.25$	0.533	0.916
$\delta = 0.50$	0.258	0.832
$\delta = 0.75$	0.084	0.710

Fig. 4. Success rate of greedy forwarding. (i) The network used in our simulation. (ii) The simulation results for varying degrees of imprecision of the node locations.

We emphasize that in all our experiments the node density is quite low relative to the communication radius (the average degree of the communication graph is always not much more than 10). This is meant to show that our method does not require extremely dense deployments, despite the fact that it relies on the fact that the nodes "sample" the plane and "witness" the various adjacencies between different regions. Sensor nodes are always deployed uniformly at random in a square, and additional holes have been created manually.

A. Success Rate of Greedy Forwarding under Imprecise Node Locations

Previously we made the claim that our method retains the benefits of the macroscopic view on network topology, e.g. robustness to small holes. As a good measure of this robustness, we considered the fraction of times when greedy forwarding alone is able to successfully deliver the message. We compared our algorithm with GPSR under various degrees of imprecision of the node locations ($\delta = 0$ denotes exact node locations, $\delta = 0.25$ that the true location of a node might be up to one quarter of the communication radius away from the location assumed by GPSR/MGGR). We obtained the results shown in Figure 4.

The simulated network had about $10,000$ nodes and an average degree of 6. In addition to being sparse, the network had many small holes. The landmarks for MGGR were chosen with $k = 5$, i.e. according to the formula mentioned above. The rates were obtained by averaging over $10,000$ trials, where in each trial a message was routed using GPSR/MGGR, between a source and a destination chosen uniformly at random. We can see that MGGR significantly outperforms GPSR, the difference becoming even more pronounced with higher degrees of imprecision of the node locations.

B. Path Length

For the setting of exact node locations we also compared the lengths of the produced paths, averaged over $1,000$ randomly chosen source-destination pairs. The network consisted of about $17,500$ nodes. We considered two different topologies on the same node distribution: a unit-disk graph (corresponding to $\alpha = 1$) and a quasi-unit-disk graph ($\alpha = 1.25$). In the first case the average degree was 10, and in the second case 8. The landmark selection parameter was $k = 5$. Figure 5 shows the network used in the simulation and the values we obtained.

Algorithm	Avg. path length	
	$\alpha = 1$	$\alpha = 1.25$
GPSR	65.99	83.40
MGGR	42.04	50.49

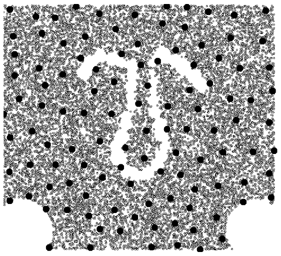
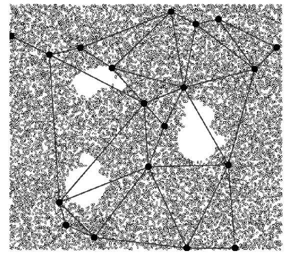


Fig. 5. Average path length. (i) The network used in simulation, and the selected landmarks. The network contains a large topological feature (the long narrow "corridor" with a small "entrance").



Algorithm	Avg. comm. cost/msg
GLIDER	257.52
GPSR	45.19
MGGR	72.39

Fig. 6. Communication cost per routed message. (i) The sparse set of landmarks used by GLIDER. (ii) Simulation results.

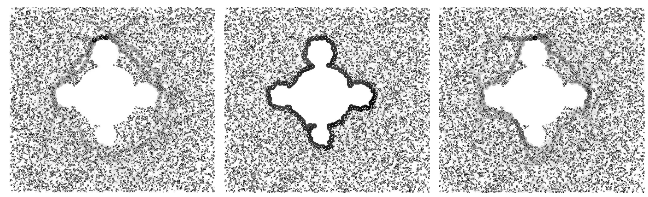


Fig. 7. Traffic load (left to right). (i) GLIDER (ii) GPSR (iii) MGGR

As expected, due to the presence of the narrow "corridor", for a significant fraction of the test instances GPSR has to invest a lot of effort in face routing. Our CDM-based algorithm is able to route more efficiently into and around the "corridor" using the landmark graph. MGGR significantly outperforms GPSR, in many cases by a factor of two or more. Also, we can see that the relative performance does not change too much between the two topologies. We did not compare directly with GLIDER but expect the latter to produce considerably better paths in this setting due to the sparsity of the network.

C. Communication Cost

In MGGR, the tile associated with each landmark is of constant size (the constant depending on the choice of the parameter k). The following experiment confirms that having small tiles reduces communication cost in the final stages of message delivery, i.e. inside the tile of the destination node. For each of $1,000$ randomly chosen source-destination pairs, we record an estimate of the number of messages needed to route one message.

We estimate the cost using the following simple model, which in our opinion provides sufficiently fair comparison. For the part of the route determined by greedy forwarding (using real coordinates or simply graph distance), we add one unit of communication cost per link traversed by the path. For the part of the route discovered by flooding a Voronoi tile, which happens in the final stages of GLIDER and MGGR, the cost is equal to twice the size of a *maximal independent set (MIS)* of this tile (we compute maximal independent sets for the tiles in advance)[5]. We felt that this is a more realistic measure than pure flooding, since most practical systems implement some simple form of scheduled broadcast with very little overhead.

In Figure 6(i) one can observe that the set of randomly chosen landmarks used by GLIDER fails to capture the holes. Figure 6 (ii) shows our results. We compared GLIDER, GPSR and MGGR, in a network of about $18,700$ nodes with average degree 10. The landmark selection parameter is $k = 5$. We found that GLIDER has to resort to flooding the last tile most of the time, i.e. using greedy routing on the the local coordinates rarely leads to the destination. Combined with the fact that the tiles are large, this leads to a poor performance by GLIDER in this metric (greedy routing on the local coordinates employed by GLIDER typically requires a higher density of the communication graph).

[5]This is in fact the size of a *connected dominating set (CDS)* obtained by adding nodes to the chosen MIS to make it connected. The size of any CDS is an upper bound on the optimal number of messages needed for a broadcast within the tile. Indeed, if only the nodes in the CDS send one message each, all other nodes will get the message.

D. Traffic Load

We also evaluated the load-balancing properties of our scheme. The network had $8,800$ nodes with average degree 9, and we again tested on $1,000$ randomly chosen pairs. The landmark selection parameter was $k = 5$. We route one message per pair and add one unit of load to each node on the path.

Figure 7 shows the visualization of the results. Darker disks represent nodes with higher load. As expected, GPSR suffers from the "hole-hugging" phenomenon, whereas GLIDER succeeds in taking the load further from the hole boundary.

We notice that MGGR does not quite match GLIDER's performance. This is due to the fact that in GLIDER a landmark can often be far from hole boundaries, and it is easy to see that the load balancing effect grows with this distance. One reason is that the landmarks tend to attract messages from the neighboring tiles, thus pulling them further into the interior. In that sense, as noted previously, by changing the tile size and the landmark separation, one can interpolate between load balancing benefits of GLIDER, and the energy efficiency of landmark-based methods.

E. Summary

In conclusion, our experiments show that the MGGR algorithm retains most of the advantages of landmark-based methods, while improving their energy efficiency and scalability. In particular, we showed that it successfully avoids small holes and does not closely follow the boundary edges, which is one of the main drawbacks of GPSR. One big improvement with respect to these methods, as the comparison with GLIDER shows, is in terms of energy efficiency, since the final intra-tile phase of message delivery only requires flooding a constant-sizes neighborhood.

ACKNOWLEDGMENTS

Both authors were supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the German Federal Ministry of Education and Research (FKZ 01IMC01).

REFERENCES

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[2] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," in *IEEE Int. Conf. on Distributed Computing and Systems Workshops; Cluster Computing*, 2001.

[3] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, 2000, pp. 243–254.

[4] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proc. MobiHoc*. New York, NY, USA: ACM Press, 2003, pp. 267–278.

[5] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: of theory and practice," in *PODC*, 2003, pp. 63–72.

[6] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *ACM DIALM*, 2002, pp. 24–33.

[7] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks," in *2nd DIALM-POMC*, 2004.

[8] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *ACM MobiCom*, 2003.

[9] S. Funke and N. Milosavljević, "Network Sketching or: "How much Geometry hides in Connectivity ? – Part II"," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[10] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang, "GLIDER: Gradient Landmark-Based Distributed Routing for Sensor Networks," in *INFOCOM*, 2005.

[11] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1-3, pp. 165–177, 1990.

[12] F. Kuhn and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proc. DIALM-POMC*, 2003, pp. 69–78.

[13] G. Carlsson and V. de Silva, "Topological approximation by small simplicial complexes," preprint, 2003.

[14] M. Erwig, "The graph voronoi diagram with applications," *Networks*, vol. 36, no. 3, pp. 156–163, 2000.

[15] A. Okabe, B. Boots, and K. Sugihara, *Spatial tessellations: concepts and applications of Voronoi diagrams*, 1992.

[16] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks." in *INFOCOM*, 2002.

[17] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, Boston, Massachusetts, USA:USENIX*, May 2005.

[18] X. Li, G. Calinescu, and P. Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *IEEE INFOCOM*, 2002.