GLIDER: Gradient Landmark-Based Distributed Routing for Sensor Networks

Qing Fang* Jie Gao† Leonidas J. Guibas† Vin de Silva‡ Li Zhang§

* Department of Electrical Engineering, Stanford University. jqfang@stanford.edu
† Department of Computer Science, Stanford University. {jgao,guibas}@cs.stanford.edu
‡ Department of Mathematics, Stanford University. silva@math.stanford.edu
§ Information Dynamics Lab, HP Labs. l.zhang@hp.com

***Abstract*— We present Gradient Landmark-Based Distributed Routing (GLIDER), a novel naming/addressing scheme and associated routing algorithm, for a network of wireless communicating nodes. We assume that the nodes are fixed (though their geographic locations are not necessarily known), and that each node can communicate wirelessly with some of its geographic neighbors—a common scenario in sensor networks. We develop a protocol which in a preprocessing phase discovers the global topology of the sensor field and, as a byproduct, partitions the nodes into routable tiles—regions where the node placement is sufficiently dense and regular that local greedy methods can work well. Such global topology includes not just connectivity but also higher order topological features, such as the presence of holes. We address each node by the name of the tile containing it and a set of local coordinates derived from connectivity graph distances between the node and certain landmark nodes associated with its own and neighboring tiles. We use the tile adjacency graph for global route planning and the local coordinates for realizing actual inter- and intra-tile routes. We show that efficient load-balanced global routing can be implemented quite simply using such a scheme.**

Keywords: Graph theory, System Design, Combinatorics, Algebraic Topology, Topology Discovery, Landmark Routing

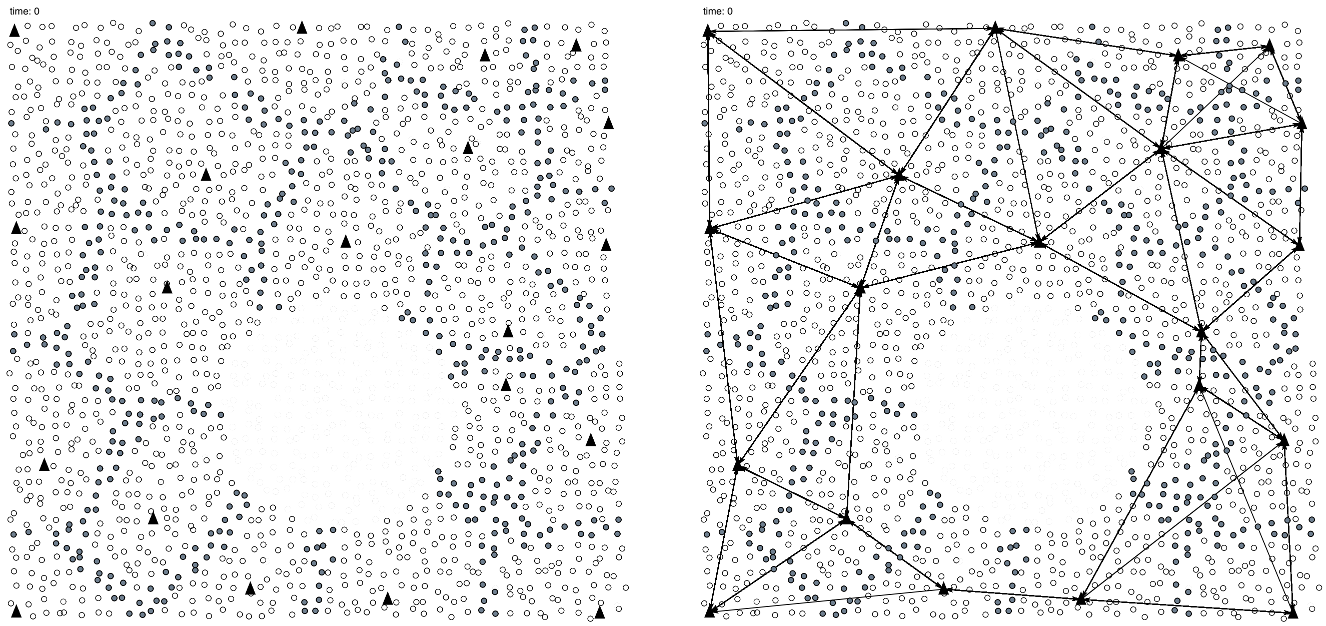
I. BACKGROUND

Techniques for routing information are central to all communication networks. Routing algorithms are intimately coupled to the way that nodes in the network are addressed or named. Such algorithms fall somewhere in the spectrum from *proactive* to *reactive* [15], according to the extent of precomputation done to facilitate route discovery. In stable networks with powerful nodes, such as the Internet, routing tables in special router nodes are proactively maintained and take advantage of the hierarchical structure of IP addresses to enable route discovery. At the other end, in *ad hoc* sensor and communication networks, where topology changes are frequent and node hardware less powerful, reactive protocols that discover a route on-demand become desirable. Unfortunately, in the absence of auxiliary data structures, reactive protocols such as AODV [12] or DSR [8], may resort to flooding the network in order to discover the desired route.

In this paper we are primarily interested in routing on wireless sensor networks. Such networks are often deployed in settings where the nodes operate untethered; thus power conservation becomes a serious concern and flooding is undesirable. Early uses of sensor networks were primarily data collection applications, requiring the one-time construction of aggregation or broadcast trees. As the sophistication of sensor network applications increases, however, there is more demand for point-to-point routing of information to support data centric storage [14] and more complex database-like queries and operations. Examples include multi-resolution storage, range searching, and the like. A survey of networking and data storage techniques for sensor networks is given in [20]. While the fragile link structure and meager node hardware of sensor networks suggests the use of reactive routing protocols, the energy overhead of flooding for route discovery can be significant and needs to be mitigated whenever possible.

One such situation is when the geographic locations of sensor nodes are known. In that case, greedy geographical routing protocols can be used in which a packet starts at the source node and is then successively relayed through other nodes to its destination with as few intermediate states as possible. At each step, the node currently holding the packet simply forwards it to the node, among its one-hop communication neighbors, which is closest to the destination. Various meanings of ‘closest’ are possible. The presence of holes in the sensor field can cause these greedy methods to get stuck in local minima; however, a variety of methods have been proposed for overcoming this difficulty and guaranteeing packet delivery, if at all possible. Probably the best known among these, GPSR [9] builds a planar subgraph of the connectivity graph and uses perimeter forwarding when greedy forwarding gets stuck. The beauty of these geographic forwarding methods is that they compute routes that are often close to the best possible, and do so with very little overhead in maintaining auxiliary routing structures. Effectively the location of a node becomes its name or address, and each node needs only to know the locations of its neighbors and that of the destination in order to decide how to forward a packet. Euclidean coordinates encode the global state and hence such algorithms can operate effectively using information which is purely local.

Although geographical location gives the nodes natural names and enables efficient routing, it is in many cases difficult or expensive to obtain accurately. GPS receivers can be costly and lead to cumbersome node form factors; furthermore, they do not work indoors, or under heavy foliage, etc. As a consequence, in most settings, it is only feasible to have a few nodes equipped with a GPS receiver. Various localization algorithms have been developed [16], [17] and must be invoked to localize the rest of the nodes. In these methods, the geographic location of a set of anchor nodes is assumed



(i) (ii)

Fig. 1. Landmarks are shown by triangles. Sensor nodes are shown as small circles. The nodes are divided into tiles. The dark nodes are the boundaries of the tiles. (i) The landmark Voronoi complex; (ii) The combinatorial Delaunay triangulation.

to be known, either manually or through GPS. Other nodes determine their location by estimating their distances to three or more of these anchors and then become anchors themselves; and so on. However, such localization algorithms are still quite expensive in terms of computation or communication, and often insufficiently accurate. Unfortunately, these inaccuracies can have deleterious effects on routing algorithms based on location information [18].

The idea of geographic forwarding is so compelling that a number of authors have tried to use geographic coordinates even when actual node locations are not available. The idea is to produce *virtual node coordinates* on which to use protocols such as GPSR. These are obtained by embedding the link connectivity graph of the nodes in the plane [10], [11], [13] so that nodes that can communicate directly are embedded near each other and those that do not are further away. Unfortunately such global embeddings can be time-consuming to compute and may not reflect well the actual geometry of the node layout. For example, in the presence of communication obstacles (such as walls), nodes that are geographically close may actually be distant in the communication graph. Also, when the actual node deployment is in 3-D, as in monitoring buildings, forcing a 2-D layout will cause large distortions with the consequence that the planarization required by GPSR and related protocols will necessarily ignore much of the actual connectivity present.

II. Topology-Enabled Routing

We present a novel routing scheme, named GLIDER, that, like the virtual coordinate schemes above, depends only on node connectivity and not on any knowledge of node positions. The key idea is to divide the problem into a global preprocessing step and a local routing problem (for which we present a specific solution). In the preprocessing step we discover the global topology of the sensor field. This gives us information about connected components and holes in the sensor field layout. In the process we partition the field into tiles. We regard these as having trivial topology, so that greedy forwarding methods based on local coordinates are likely to work well within each tile.

Our intuition is that, in many of the real-world situations where sensor networks may be deployed, the topological features of the layout (e.g. holes) will be few and will mostly reflect the underlying structure of the environment (e.g. obstacles). Moreover, this relatively simple global topology is likely to remain stable: nodes may come and go, but such changes are unlikely to destroy or create large-scale topological features. It follows, if the global topology is stable, that we can afford to carry out proactive routing at an abstract combinatorial level. These high-level routes can then be realized as actual paths in the network by using reactive protocols.

For example, the node distribution shown in Figure 1 has a large hole in the middle. If two nodes are situated on opposite sides of the hole, there are two ways of reaching one node from the other: clockwise and counter-clockwise around the hole. This is a *topological* statement. Having made the topological decision whether to go clockwise or counter-clockwise, we can use local decisions to select the specific path on a node-by-node basis. In this way, we have broken the routing problem into two phases: a global planning phase, in which the combinatorial structure of the path is determined using global information; and a local phase, in which the combinatorial path is implemented as an actual sequence of hops, selected using local greedy methods.

For this second phase, we define sets of local coordinates that depend only on the link connectivity of the nodes. Gradient descent on these coordinates will naturally follow local geodesics of the layout. However, our procedure is less sensitive to sensor geometry than is classical geographic routing. For example, suppose we have a collection of sensors densely deployed in two big rooms connected by a long narrow corridor, as shown in Figure 2. The geometry of the sensor field is curved, so greedy geographic routing between the rooms will almost certainly fail if it is based directly on the coordinate system implied by the diagram. On the other hand, the topology of this sensor field is fundamentally the same as the topology of an array of sensors deployed densely inside a convex shape such as a disk. Our global-local scheme will thread the route through the corridor without difficulty. This works because we limit the local greedy processes to certain regions (the tiles) whose topological structure is known to be sufficiently nice. Our knowledge of the global network topology allows our routing scheme to avoid some of the more common pitfalls and limitations of current global coordinate-based schemes; such as the limitation to two dimensions, or the need to construct a planar graph to bypass local minima [9], [1], or the explicit discovery of holes [5].

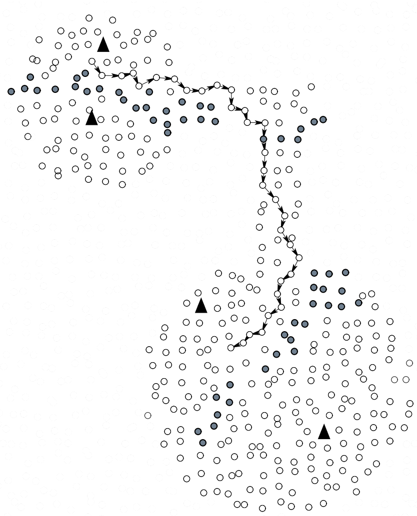


Fig. 2. A narrow corridor connects two rooms. GLIDER discovers a route that goes through the corridor, following naturally defined gradients.

Both phases of our algorithm—global topology discovery and local coordinates—are based on the selection of an appropriate subset of the nodes designated as *landmarks*. We use combinatorial Voronoi/Delaunay techniques to extract a topological complex whose vertices are the landmarks and whose topology captures the topology of the underlying sensor field. At the same time, we generate a set of local coordinates for each node which are derived from the node's link distances to nearby landmarks. These coordinates are easy to generate since they depend only on local information—we make no attempt to provide a global geometric embedding for the entire network.

The idea of using landmarks for routing is of course not new to the networking community. Tsuchiya [19] proposed a hierarchical landmark-based scheme for generating node names or addresses. Our use of landmarks for addressing and route generation is quite different—we use landmarks to partition the sensor field into geographically routable tiles. Recently, we became aware of another paper currently under review that uses distances to landmarks as coordinates [7] and provides extensive simulation results. Our work differs in that we do not use all the landmarks to provide coordinates for all the nodes. It follows that our scheme scales better to large networks. Moreover, we have a different method for generating the virtual local coordinate systems, which is guaranteed to route correctly in the continuous domain and which empirically works well in practice.

In a quite different subject area, nonlinear dimensionality reduction (NLDR), landmarking techniques were introduced in [4] to simplify expensive calculations by using a sparse approximate representation of the global geometry of a data set. The goal in NLDR is to find explicit low-dimensional coordinates for viewing high-dimensional nonlinear data. The local landmark coordinates used by GLIDER are closely related to the coordinate embedding functions derived in [4].

As an aside, it is somewhat unfortunate that the current usage of the term *topology discovery* in the networking community refers only to the discovery of local link relationships between nodes and to the lowest-order topological invariant, namely path connectivity. Our use of the term topology discovery in this paper refers more broadly to an understanding of the global topology of the sensor field in the sense of algebraic topology; for example, we consider higher order topological features such as holes in 2-D, tunnels and voids in 3-D, and so on.

We note that in traditional algebraic topology the objects of study are continuous spaces rather than discrete collections of points or nodes. That being so, when we talk about the topology of a finite set of points sampled from an underlying object, we really mean the topology of the underlying object itself. To recover this from the points alone, we can conceptually transform the discrete cloud of points into a continuous space by putting a small ball around each point, making sure that the balls associated to nearby points have sufficient overlap. From this one can build discrete structures, namely simplicial complexes, that provably capture the topology of the underlying continuous object [2] under mild conditions. The same idea can be applied to the task of understanding the topology of a field of communication nodes. However, the lack of positional information, the need to minimize computational costs, and the desire to perform topology estimation in a distributed way, all make the network case more challenging.

III. Overview of Glider

As mentioned earlier, our scheme for route discovery and node naming/addressing is based purely on link connectivity information, and it works by separating the global topology and the local connectivity. Formally, suppose that $G = (V, E)$ is a communication graph on the sensor nodes V. The edges E are unweighted: they identify which pairs of nodes have direct communication but not the geometric distance between those nodes. The *graph distance* between two nodes is simply the number of edges (or hop count) in the shortest path between them.

Given the graph G, we assign a *name* to each node in V. We also construct an auxiliary atlas $M(G)$ which is shared by all the nodes. A *local name-based route discovery* scheme is a relay scheme which functions as follows. For any destination v specified by name, and for any node u, the scheme specifies a successor node chosen from the neighbors of u. By jumping repeatedly from node to successor, the destination v is eventually reached. The choice of successor depends only on the names of v, u and the neighbors of u, and on the auxiliary atlas M.

An alternative view is that the communication graph G is decomposed into two parts: the common auxiliary atlas M which encodes global connectivity information that is accessible to each node, and the node names which encode node specific information stored distributedly in each node. In a trivial way, one can simply let M equal G or (in the other extreme) arrange for each name to encode the entire communication graph and the position of the node. Our goal is to reduce the size of M and the length of names by exploiting the fact that G is a communication graph of sensors deployed within some geometric space.

To compute the auxiliary atlas M, we estimate the global topology of the sensor field by partitioning the nodes into *routable tiles* and extracting the adjacency relations between these tiles. The goal is for each routable tile to have trivial topology, so that simple greedy routing will work well within the tile. Meanwhile, the global connectivity structure of the set of tiles provides a compact high-level atlas of the sensor field. Our particular partition is defined by selecting a small set of well-dispersed nodes to be *landmarks*, and letting the tiles be the Voronoi cells of the landmarks, where the Voronoi cell of a landmark u is the set of nodes whose nearest landmark is u (in the hop-count metric). Ties are permitted, so a node may belong to more than one tile. The cell complex associated to such a partition is called the *landmark Voronoi complex* (LVC). The dual complex of the LVC has been called the *combinatorial Delaunay triangulation* (CDT) [3]. It is this which serves as our auxiliary atlas M. The details of constructing LVC and CDT are described in Section IV.

The name of each node consists of two parts: the *global tile name* and the *local landmark coordinates*. The global tile name of a node is simply the identity (unique ID) of its closest landmark; this identifies the tile containing the node. (If the node belongs to more than one tile, one can be chosen arbitrarily.) The local landmark coordinates are derived from the set of distances from the node to its nearby landmarks. Specifically, we use 'centered squared-distance coordinates,' which we describe in Section V. It turns out that gradient descent on the Euclidean distance function in these coordinates gives an effective greedy routing algorithm. More precisely, in the continuous domain we can prove under mild conditions that this algorithm always succeeds. In the discrete case, our experiments show that this scheme has high success rate even for sparse sensor deployment. In Section V, we describe the local landmark coordinate system in detail.

In summary, the preprocessing phase discovers the global topology by building the landmark Voronoi complex, and constructs the local coordinate system for each tile. Every node is given a name reflecting these components; and every node has knowledge of the combinatorial Delaunay triangulation, which captures the global topology of the sensor field in a compact lightweight structure. When a node is presented with a routing request, it first calculates from the combinatorial Delaunay triangulation a sequence of tiles for the routing path. Then, to select the next node in the route, the node uses greedy gradient descent towards the next tile in the path, or towards the final destination (if the final tile has been reached). The details are given in Section VII-B.

The success of our approach depends on making a reasonable choice for the set of landmarks. We discuss this further in Section IV-B. In many common situations we can expect that the complexity of the topological features of our complex will reflect the complexity of the topological features of the environment in which the sensor nodes are deployed, such as physical obstacles that prevent node placement. We expect these to be large-scale features and few in number. As a consequence, the number of landmark nodes needed will also be small—as this number is proportional to the topological complexity of the field. Thus the combinatorial Delaunay complex is a small structure and it is reasonable to assume that it can be stored at, or easily accessible from, every node.

IV. Landmark Voronoi Complex (LVC)

For a set of nodes V and a communication graph G, the landmark Voronoi complex captures the global topology of the network using only the local link connectivity. We may assume that G is connected, since we can otherwise just consider each connected component separately. We denote by $\tau(u, v)$ the topological length (hop count) of the shortest path between u, v in the communication graph.

A. Definition

The landmark Voronoi and Delaunay complexes are the natural extension of the geometric Voronoi diagram, and its dual Delaunay triangulation, to the case of a graph with the shortest-path metric. For a graph $G = (V, E)$ and a subset of landmarks $L \subset V$, define the Voronoi cell $T(v)$ of a node $v \in L$ to be the set of nodes whose nearest landmark is v (ties are allowed). See Figure 1(i). Formally:

$$T(v) = \{u \in V \mid \forall w \in L\,,\, \tau(u, v) \leq \tau(u, w)\}$$

We note the following property of a Voronoi cell.

Lemma 1. *For any node $u \in T(v)$, the shortest path from u to v is completely contained in $T(v)$.*

Proof. If the lemma were false, there would exist $w \notin T(v)$ on the shortest path from u to v. Since $w \notin T(v)$, there exists $x \in L$ such that $\tau(w,x) < \tau(w,v)$. Thus, $\tau(u,x) \leq \tau(u,w) + \tau(w,x) < \tau(u,w) + \tau(w,v) = \tau(u,v)$. This contradicts the hypothesis that $u \in T(v)$; so the lemma must be true. □

One implication of this lemma is that the spanning graph on each Voronoi cell is connected. Thus, the Voronoi cells of a set of landmarks provide a natural partitioning of the sensor field into *connected* tiles. A stronger requirement is that the tiles have trivial topology in all dimensions (not just connectivity). When the sensor field has large holes, we find that appropriately-chosen landmarks can effectively fragment the sensor field into subsets with simple topology. See Figure 1(i).

The Voronoi cells form the *landmark Voronoi complex* (LVC). Following [3], we use a dual *combinatorial Delaunay triangulation* (CDT) to record the adjacency relation between Voronoi cells.[1] For our purposes, the combinatorial Delaunay triangulation $\mathcal{D}(L)$ is a modified dual of the LVC, defined as follows. Write $w \sim w'$ to mean that nodes w, w' share an edge in G or are the same node. Then the vertices $v_1, \dots, v_k$ span a simplex in $\mathcal{D}(L)$ if and only if there exist nodes $w_1, \dots, w_k$ such that $w_i \in T(v_i)$ for all i and $w_i \sim w_j$ for all i, j.

Under favorable conditions (a dense distribution of nodes, reasonably simple topology, a small number of well-separated landmarks) suitable variations of the LVC and CDT complexes successfully capture the global topology of the communication network [3]. These constructions give us the potential to detect and exploit high-order topological characters of a sensor field. Having said that, in this paper we are mainly interested in the connectivity graph $D(L)$ of the landmarks, i.e. the 1-dimensional skeleton of CDT (Figure 1(ii)). The edge v_1v_2 belongs to $D(L)$ iff there exist nodes w_1, w_2 with $w_1 \sim w_2$ and $w_i \in T(v_i)$ for $i = 1, 2$. The nodes w_i are referred to as *witnesses* to the edge v_1v_2.

For connectivity, we have the following easy result:

Theorem 2. *If G is connected, then the combinatorial Delaunay graph $D(L)$ for any subset of landmarks L is also connected.*

Proof. We must show that there is a path in $D(L)$ between any pair of landmarks u, v. Since G is connected, there is certainly a path from u to v within G. Let us suppose this path visits nodes $w_0, w_1, \dots, w_k$ in sequence, where $w_0 = u$ and $w_k = v$. Each node w_i belongs to some Voronoi cell $T(x_i)$ where each $x_i \in L$. We may assume that $x_0 = w_0 = u$ and $x_k = w_k = v$. We claim that the sequence $x_0, x_1, \dots, x_k$ represents a valid path in $D(L)$ from u to v. Specifically, for each $0 \leq i < k$, we claim that $x_i \sim x_{i+1}$ in the graph $D(L)$. This is clear, since w_i, w_{i+1} are witnesses for the edge x_ix_{i+1} in the case that $x_i \neq x_{i+1}$. □

[1] Our definitions diverge slightly from the definitions in [3]. The problem in both cases is to ensure that CDT maintains the connectivity of the original graph; something that is not quite true with the natural definitions. We deal with this in a different way than [3].

The proof of Theorem 2 amounts to the stronger assertion that every path in G can be 'lifted' to a path in $D(L)$. Conversely, every path in $D(L)$ can be realized as a path in G. This follows from the case of a single edge $v_1v_2 \in D(L)$. Let w_1, w_2 be witnesses for v_1v_2. For each i the shortest path from v_i to w_i lies entirely within $T(v_i)$, by Lemma 1. We can concatenate these paths to obtain a path $v_1 \dots w_1w_2 \dots v_2$ in G which lifts to the length-1 path v_1v_2 in $D(L)$.

These last assertions provide strong corroboration of one of the main claims in this paper, which is that the CDT graph is an appropriate simplification of the communication graph G for determining a global routing strategy.

We summarize the main points of this section: For a set of chosen landmark nodes, the Voronoi cells of the landmarks provide a partitioning of the sensors. Each Voronoi cell is connected and has 'simple' topology when the landmarks are well-chosen. The combinatorial Delaunay triangulation encodes adjacency information between Voronoi cells, and provides a compact high-level atlas for the sensor field which is suitable for global route-planning.

B. Landmark selection

While the definition and properties of LVC and CDT hold for any subset of landmarks, careful selection of landmarks is crucial for the effectiveness and efficiency of routing. Since CDT serves as the auxiliary atlas provided to every node, it should be as small as possible so that it can be replicated in the network with minimal cost. On the other hand, we need enough landmarks to ensure that each Voronoi cell has simple (i.e. 'hole-free') topology. These are the two opposing goals.

The landmark selection problem bears some resemblance to the sampling problem for mesh generation—we particularly desire to have several landmarks lying close to topological features, such as hole boundaries. Hand-picked landmarks are one option, since in many cases the presence of holes may be known *a priori* to those deploying the network. It is also possible to automatically discover hole-boundaries [6], or at any rate a few nodes on the boundary [13]. With such information, we can arrange for nodes near the boundary to be selected as landmarks with higher probability than interior nodes. In general we expect the number of landmarks to be proportional to the number of holes (or topological features) of the sensor domain. We are usually interested in domains with a small number of large holes, in which case the landmark set and hence the CDT complex will be small enough to distribute to the entire set of nodes.

V. Local Landmark Coordinates

Under ideal circumstances with well-chosen landmarks, the nodes in each cell of the landmark Voronoi complex will be nicely distributed. By this we mean that the shortest-distance metric in each tile approximates the metric in a finite sample of a convex Euclidean region. One general strategy for routing on such network is to supply a coordinate system to the nodes, and then perform greedy routing by forwarding packets to a neighbor which is closer to the destination according to these coordinates. The use of the Euclidean coordinates of the sensors is one natural choice but these coordinates may be difficult or expensive to obtain. Here we propose a virtual coordinate system which is easy to compute, is guaranteed to be free of local minima in the continuous plane, and which in practice works well in the discrete case. These are the local landmark coordinates. We first describe these coordinates in continuous Euclidean space, and then extend the definitions to the discrete case.

A. Continuous version

It is easiest to understand our coordinate system (we will define it shortly) in the continuous case. The goal is to construct a set of coordinate functions depending only on the distances to some fixed set of landmark points, in such a way that gradient descent on the distance function to a target point always reaches the target successfully. In other words, the distance function should have no local minima other than the global minimum.

Let $\{u_i\}$ be a set of k landmarks in the plane. The natural first guess is to assign to each point p the virtual coordinate vector $A(p) = (|p - u_1|, |p - u_2|, \ldots, |p - u_k|)$, where $|p - u_i|$ is the Euclidean distance between p and u_i. The virtual distance in this coordinate system between points p, q is then $d(p,q) = |A(p) - A(q)|^2 = \sum_{i=1}^{k}(|p-u_i| - |q-u_i|)^2$. Given a destination q, the greedy routing algorithm operates by gradient descent on this function with respect to p.

There are simple examples with three landmarks which show that this process can get stuck in local minima. One can do slightly better with the squared-distance vector $B(p) = (|p-u_1|^2, |p-u_2|^2, \ldots, |p-u_k|^2)$. It can be shown that there are no local minima when $3 \le k \le 9$ and when the destination is inside the landmark convex hull. When $k > 9$ or when the destination is outside the convex hull, there is no such guarantee. Figure 3(i) shows an example where the gradient flow can get trapped in a local minimum. For this reason we introduce *centered landmark-distance coordinates* $C(p)$. The i-th coordinate is defined by $[C(p)]_i = [B(p)]_i - \bar{B}(p)$, where $\bar{B}(p)$ is the mean of the entries of $B(p)$. The modified virtual-distance function is then $d(p,q) = |C(p) - C(q)|^2$. The advantage of this is made clear by the following lemma.

Lemma 3. *In the continuous Euclidean plane, gradient descent on the function $p \mapsto d(p,q)$ always converges to the target q, provided that there are at least three non-collinear landmarks.*

Proof. We can explicitly evaluate

$$[B(p)]_i = |p|^2 - 2p \cdot u_i + |u_i|^2 \,,$$

and hence

$$[C(p)]_i = -2p \cdot (u_i - \bar{u}) + w_i \,,$$

where $\bar{u} = \frac{1}{k}\sum_j u_j$ and $w_i = |u_i|^2 - \frac{1}{k}\sum_j |u_j|^2$.

The function $p \mapsto C(p)$ is therefore an affine linear transformation. Under the assumption that there are at least three non-collinear landmarks, we now show that the map is one-to-one. The idea is to find at least one point in the plane which is determined uniquely by its coordinates, because then (for an affine map) the same must be true for all points in the plane. The circumcenter of any three non-collinear landmarks is such a point, since it is uniquely determined by the property that the corresponding three coordinates are equal. This establishes that the map is one-to-one, in addition to being affine linear. It follows that gradient of the distance function is nowhere zero except at the destination itself. □

In k-dimensional Euclidean space, the minimum requirement is $k+1$ landmarks not contained in any $k-1$-dimensional affine subspace. In other words, the affine span of the landmarks must be the entire k-space.

We note that the straight line path to the target is a descending trajectory for the distance function d. In general it is not the path of steepest descent. Figure 3(ii) shows the same configuration as in Figure 3(i), but with the distance to target measured in the centered landmark-distance coordinates. In that case there is no local minimum.

B. Discrete version

In a graph setting, the discrete version of the greedy routing algorithm uses hop counts to the landmarks as a replacement for Euclidean distances. In situations where the nodes are densely distributed, the minimum number of hops to a landmark is a fair approximation to the Euclidean distance to that landmark.

For a set of landmarks $\{u_1, u_2, \ldots, u_k\}$ and for any node p, let $\tau(p, u_i)$ denote the graph distance (i.e. the minimal hop count) between p and u_i. Let $\bar{\tau}(p) = \sum_{i=1}^{k} \tau(p,u_i)^2/k$. We then assign to p the centered virtual coordinate vector

$$C(p) = (\tau(p,u_1)^2 - \bar{\tau}(p), \ldots, \tau(p,u_k)^2 - \bar{\tau}(p)) \,.$$

The centered virtual distance between two points p, q is then $d(p,q) = |C(p) - C(q)|^2$ just as in the continuous Euclidean case. Given a destination q, our greedy routing algorithm chooses the neighbor r of p which minimizes $d(r,q)$. In other words, we move packets by greedy minimization of the Euclidean distance to the target, measured in the virtual coordinate system. This algorithm is local and efficient since only the virtual coordinates of the neighbor nodes are needed.

In the discrete version, we can no longer guarantee that local minima do not exist. A packet may hit a node for which all the neighbor nodes have virtual distances further away from the destination. However, when the nodes are dense enough, the shortest distance metric approximates the Euclidean metric closely enough to reduce the chance of local minima. Another cause of instability is the eccentricity of the affine transformation in Lemma 3. When the landmarks are nearly collinear—e.g. when $\tau(u,v) + \tau(v,w) \approx \tau(u,w)$ with three landmarks u, v, w—the gradient field in the continuous case is quite shallow in certain directions. This can be seen quite clearly in Figure 3(ii). Under these circumstances the discrete approximation is more likely to suffer from local minima.

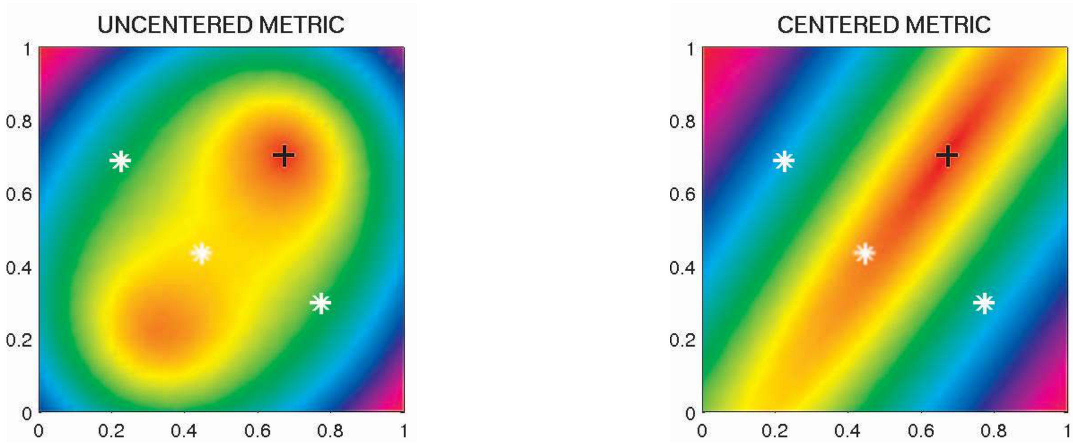


Fig. 3. The distance function for landmark-based greedy routing. There are 3 landmarks marked by snowflakes. The destination is marked by a + sign. The color of a point represents its distance to the destination with respect to (i) uncentered coordinates; (ii) centered coordinates. Note the local minimum in the uncentered case.

Local landmark coordinates are only used for routing between nodes in the same tile, so we can define these coordinates in terms of a small set of nearby landmarks called *reference landmarks*. Specifically, for the nodes in the Voronoi cell $T(v)$ of a landmark v, the reference landmarks are v itself and the neighbors of v in $D(L)$. For technical reasons, when we compute distances between each node and its reference landmarks we do not use shortest paths within G; instead we define a *neighborhood distance* metric as follows. For each landmark $v \in L$, its *Voronoi neighborhood* is defined as $U(v) = T(v) \cup \bigcup_{(u,v)\in D(L)} T(u)$, i.e. as the union of the Voronoi cells of v and its neighbors. By Lemma 1, $U(v)$ is connected. For a landmark v and a node $u \in U(v)$, their neighborhood distance is defined as the graph distance from u to v measured in the subgraph spanned by $U(v)$.

We refer to the resulting (centered squared neighborhood distance) coordinates as *local landmark coordinates*.

VI. Naming and Routing

In this section, we describe the naming and routing scheme in GLIDER by using the landmark Voronoi complex and local landmark coordinates. We will describe our implementation of the scheme in Section VII.

A. Naming of nodes

We distinguish the ID and the name of a node. The ID of a node is a number or string that uniquely identifies a node. It is usually assigned to each node before the network is formed, e.g. when the nodes are manufactured. A node's *name* is assigned after preprocessing, and it depends on the connectivity of the network and other information such as the choice of landmarks. The name is used to distinguish and address network nodes for the purpose of routing or other higher-level applications such as information gathering. Usually the node names need not be unique, provided that ambiguity can be quickly resolved—e.g. if nodes with the same name are within close vicinity.

In GLIDER, once the Voronoi complex is constructed, each node belongs to a Voronoi cell. We call that cell the *resident tile* of the node, and we call its landmark the *home landmark*. The name of a node includes the ID of its home landmark. In addition, each node includes the list of neighborhood distances to its reference landmarks, as defined in Section V-B. For a node v, let $h(v)$ denote the name of its home landmark, and $A(v)$ its vector of neighborhood distances. As defined earlier, the local landmark coordinate vector $C(v)$ is obtained from $A(v)$ by squaring and centering.

We note that witness nodes may belong to multiple Voronoi cells. In that case the home landmark may be chosen by breaking the tie in some arbitrary manner; for example, by assigning a linear order to the landmark IDs and picking the landmark with the smallest ID among the valid candidates. A more serious problem in the discrete domain is that the local landmark coordinates may not uniquely specify the node. In practice this ambiguity happens rarely provided that each node has several reference landmarks and enough neighbors. Even when this ambiguity arises, nodes with the same coordinates are likely to be close to each other, in which case a local flooding can easily resolve the situation. We also comment that, in typical data-centric applications built on sensor networks, it may be even less of a problem, since nearby nodes often carry similar data, and may not need to be distinguished if they are close enough.

B. Routing

Suppose that we wish to route from node u to node v. Routing in GLIDER consists of two stages: global routing and local routing.

• **Global routing**. This amounts to identifying the shortest path from $h(u)$ to $h(v)$ in CDT. It can be done by a look-up in the precomputed shortest-path tree rooted at $h(u)$. The path provides a sequence of tiles for the journey; say $T_1, T_2, \ldots, T_k$ where $T_i = T(u_i)$ for landmarks u_i, with $u_1 = h(u)$, $u_k = h(v)$.

• **Local routing**. Local routing consists of *inter-tile routing*, responsible for discovering path from tile T_i to T_{i+1}, and *intra-tile routing*, responsible for discovering the path towards v once T_k is reached.

Intra-tile routing is done by gradient descent using the local landmark coordinates. More specifically, once the packet reaches a node $w \in T(h(v))$, it is relayed to a neighbor which is closer to the landmark v in the Euclidean distance on the local landmark coordinates. If such a node does not exist, i.e. if a local minimum is reached, then the fail-safe option is to initiate flooding within the tile.

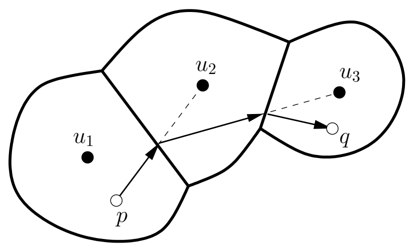


Fig. 4. Routing across tiles.

According to the discussion in Section IV, for any $p \in T_i$ there exists a path from p to T_{i+1} in the Voronoi neighborhood of u_{i+1}. Inter-tile routing is responsible for discovering such a path. Since T_i and T_{i+1} are adjacent, p lies inside the Voronoi neighborhood of u_{i+1}. We can forward the packet towards u_{i+1} by picking a node which decreases the neighborhood distance to u_{i+1}. Eventually the packet must reach the tile T_{i+1}, at which point we switch the temporary destination to u_{i+2}; and so on until the packet reaches reaches $T(h(v))$ (Figure 4).

The name of protocol, GLIDER, is meant to capture this notion of gliding down a sequence of potential slopes during this trip to the destination.

VII. IMPLEMENTATION

There are two phases in GLIDER: topology discovery and routing. Corresponding to the two phases, we introduce the *naming protocol* and the *routing protocol*.

A. Naming protocol

The topology discovery phase begins after the landmarks are selected. The naming protocol is designed to carry out the following tasks:

1) construct the landmark Voronoi complex (LVC) in a distributed fashion;
2) compute a routing table on the graph of the combinatorial Delaunay triangulation (CDT);
3) assign to each node its local landmark distance coordinates with respect to its reference landmarks.

The topology discovery phase involves several floodings of the network. Our algorithm is designed such that the number of messages in each flooding is linear in the number of nodes in the network and is independent of the number of landmarks. This improves the scalability of GLIDER to cases where many landmarks are needed.

In the first round of flooding, we compute Voronoi cells and the graph-distances from each node to its nearest landmarks. We arrange things so that each landmark floods only a small 'neighborhood of influence' rather than flooding the entire network. More specifically, each landmark u initiates a flooding message (ID_u, ℓ) where ID_u is the ID of u and $\ell = 1$. Every node v in the network maintains a list S_v of the current closest landmarks, together with the current shortest distance τ_v. Initially, $S_v = \emptyset$ and $\tau_v = \infty$. Upon receiving a flooding message (ID_u, ℓ), there are three cases.

1) if $\ell > \tau_v$, discard the message;
2) if $\ell = \tau_v$
   - if $\mathrm{ID}_u \in S_v$, discard the message;
   - if $\mathrm{ID}_u \notin S_v$, add ID_u to S_v, and broadcast $(\mathrm{ID}_u, \ell + 1)$ to all the neighbors;
3) if $\ell < \tau_v$, set $S_v = \{\mathrm{ID}_u\}$, $\tau_v = \ell$, and broadcast $(\mathrm{ID}_u, \ell + 1)$ to all the neighbors.

If the landmarks initiate this process at approximately the same time, and each message travels at approximately the same speed, then any given landmark's flooding message will be dropped when it starts to 'penetrate' the Voronoi cells of other landmarks. This cuts down the total number of messages transmitted in the flood.

At the end of this, the list S_v contains precisely the set of landmarks that are the closest to v with common shortest distance τ_v. Equivalently, S_v is the set of landmarks u such that $v \in T(u)$.

At this point, every node knows which Voronoi tile(s) it belongs to. By consulting its neighbors, a node can now determine whether it is a witness to an edge u_1u_2 in the CDT graph. Specifically, denote by N_v the set of neighbors of v in the communication graph G. For each landmark node $u \in S_v$, node v constructs the set

$$L_v(u) = (S_v \cup (\bigcup_{w \in N_v} S_w)) \setminus \{u\} .$$

With this definition, a landmark u' belongs to $L_v(u)$ iff v is a witness for the edge uu' in CDT. If $L_v(u)$ is not empty then v sends the list $L_v(u)$ to landmark u. This is quite straightforward: messages within a tile $T(u)$ can be relayed to u by greedily reducing the graph distance to u at each stage. It follows from Lemma 1 that there is no danger of getting stuck. This concludes the distributed construction of the LVC and the CDT graph.

We now designate one node to poll the landmark nodes and collect their CDT neighbor information. Having done so, this polling node floods this information to all the landmarks. Each landmark computes the CDT shortest-path tree rooted at that landmark, and then broadcasts the tree to all the nodes inside its Voronoi cell. This shortest path tree serves as the global routing table for the nodes in the cell. In addition, each node v knows the ID of its home landmark u and also of its other reference landmarks, since these can be read off as the neighbors of u in the shortest-path tree.

The final stage is to compute, for every node v, the neighborhood distances between v and its reference landmarks. This can be achieved by initiating a new flood from each landmark u which is confined to $U(u)$. Every node knows by now whether it belongs to $U(u)$, so whenever the flooding message reaches a node outside $U(u)$ it is simply discarded. Once every node v has obtained its vector $A(v)$ of neighborhood distances to its reference landmarks, the local coordinate vectors $C(v)$ can be computed using the prescription in Section V-B.

B. Routing protocol

After successful completion of the naming protocol: (i) the global topology of the network is captured by the CDT graph; (ii) each node stores the CDT shortest-path tree rooted at its home landmark; (iii) each node stores the neighborhood distances to its reference landmarks.

The GLIDER routing protocol runs on top of this infrastructure. The header of a packet contains a 'temporary destination landmark' (TDL) bit together with a integer that saves the ID of a temporary destination landmark. When a packet and the name of its destination are received at a node v, the node determines, by comparing names, whether the destination belongs to the same tile or a different tile. For the actual forwarding process—determining which node receives the packet next—there are two scenarios to consider:

• **Intra-tile routing**. When the destination is inside the current tile, GLIDER uses the greedy routing algorithm described in Section V-B. If all the neighbors of v are further away from the destination than v itself, flooding within the tile is used to complete the delivery of the packet to the destination. Otherwise, v forwards the packet to a neighbor whose distance to the destination is least among all neighbors of v.

• **Inter-tile routing**. If the destination is not in the current tile, routing follows the method indicated in Section VI-B. The node v first checks whether the temporary destination landmark bit is set. If TDL is not set, or if TDL is set but the actual temporary destination landmark stored in the header is the home landmark of v, then v consults its landmark routing table to find the next tile u_{i+1} in a shortest-path route in CDT to the destination tile. Having done so, v sets TDL to TRUE and saves the ID of u_{i+1} in the packet header.

If TDL is set, and the indicated temporary destination u_i is not the home landmark of the current node v, then v greedily picks any of its neighbors in $U(u_i)$ which is closer than v to u_i in neighborhood distance. This is always possible since $U(u_i)$ is connected. When there are multiple such neighbors, we pick one randomly. This randomization achieves better load balancing without hurting the quality of the path.

C. Data structures

Landmarks are used as logical reference points in determining nodes' local coordinates. However, from the programming point of view, they are just ordinary nodes. No extra processing power or memory is required. The information stored at a node is shown in Figure 5.

```
Node{
    the shortest path tree on CDT rooted at its home landmark;
    neighborhood distances to its reference landmarks;
    a bit to record if the node is on the boundary of a tile;
    the IDs of its neighbors
}
```

Fig. 5. Information stored at a node.

It is apparent that the local memory required for each node scales well with network size. Except for the routing table on the landmarks, each node stores only local information. Since the number of landmarks is small (23 landmarks out of 2000 nodes in our simulations) the total memory required for each node is manageable.

The global routing table on the landmarks is stable over time. The combinatorial Delaunay triangulation is a compact structure that captures the global topology of the sensor deployment, and which only changes when a large number of nodes disappear. As in Figure 1(ii), a CDT edge at the bottom of the hole disappears only when a band of nodes die so that the two corresponding landmarks are not directly connected.

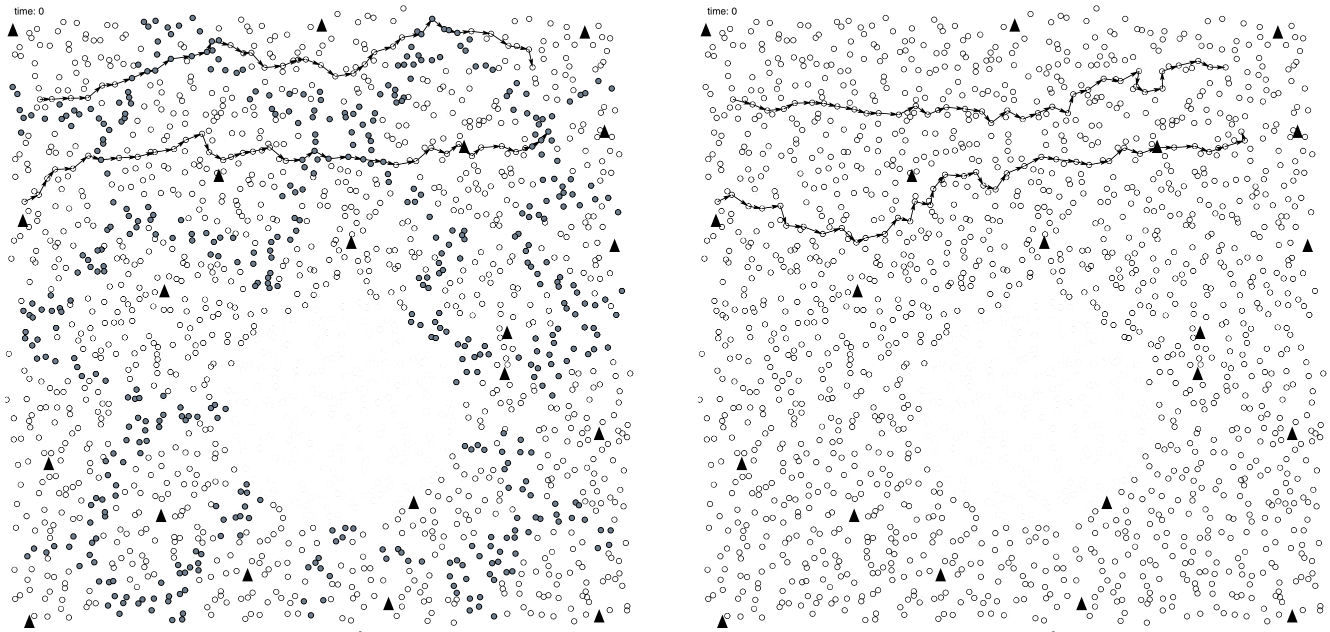
VIII. SIMULATIONS

We implemented the GLIDER protocols using C++. Although our simulations do not take into consideration typical details of network behavior—such as packet loss, packet delay and timing, and so on—these simulations empirically verify the correctness of the algorithm and the feasibility of the protocols. Network-level simulations using `ns-2` will be undertaken in the near future, in order to verify that this routing scheme is practical for real world deployment.

We simulated a network with 2000 nodes distributed on a perturbed grid. The communication graph used is the unit disk graph on the nodes. Two nodes can communicate directly if their Euclidean distance is at most 1. After the communication graph is generated, the Euclidean coordinates are discarded since our protocols use only the communication graph. Among 2000 nodes, 23 are chosen as landmarks. Of the landmarks 18 are chosen randomly, with another 5 nodes added near the network boundary, after the random selection. In all the figures, sensors are shown as small circles and landmarks are shown as larger triangles. Gray circles represent nodes on the boundaries of Voronoi cells, or equivalently the witnesses of CDT graph edges.

A. Success rate of landmark greedy algorithm

Using GLIDER, a packet can always make progress across intermediate tiles. However it may get stuck at small holes as it progresses towards the destination in the final tile. Node density is an important parameter in estimating the frequency at which a packet gets stuck. Although this work is intended for dense sensor fields with large holes in the communication



(i) (ii)

Fig. 6. A 315m by 315m region with 2000 nodes distributed on a perturbed grid. The standard deviation of the perturbation (Gaussian random variable) is equal to 50% of the radio range (11m). There are many small holes and one large hole in the network. There are a pair of source nodes on the left and a pair of destination nodes on the right. Routes between source and destination are shown as sequences of arrows. (i) Routes generated by GLIDER. Landmarks are shown as triangles. The network is divided into tiles. The darker nodes form boundaries of the tiles. (ii) Routes generated by GPSR.

average number of neighbors	2.9	3.2	4.1	≥ 5.3
percentage of success	20	70	95	100

TABLE I. The success rate of the greedy routing

graph, the routing algorithm is also quite tolerant to sparse node distribution with lots of small holes. We simulated a network with 2000 nodes distributed on a perturbed grid. The degree of perturbation is simulated using a Gaussian random variable with standard deviation equal to 50% of the radio range. We ran experiments on the success rate of greedy routing by varying the radio range (thus varying the average number of one-hop neighbors). The results are shown in Table I. For each scenario, we tried 20 pairs of sources and destinations selected at random, with path length about 40 hops in each case. The results indicate that an average of 5 or more neighbors is needed to ensure the success of greedy routing. In all the figures shown in this paper, each node has 5.3 one-hop neighbors on average

B. Load balancing and path length

In the absence of obstacles, our topology-based routing algorithm generates routes that are comparable to those generated by geographical routing algorithms. Figure 6 (i) shows two sample routes generated by GLIDER between two pairs of source and destination nodes. For comparison, routes generated by GPSR are shown in (ii). Although the actual routes generated by the two algorithms are quite different, their lengths differ by at most 2 hops out of about 40 hops in total path length.

When obstacles, i.e. holes, are present in the communication graph, geographical routing schemes either use planar graphs [9], [1] or attempt to discover hole boundaries [5] to get around holes. When a packet gets stuck, it is routed along the boundary of a hole until greedy forwarding becomes possible again. Paths that are routed around the same hole tend to merge at the hole boundary. Figure 7 shows an example. This traffic pattern is an inherent consequence of the combination of greedy forwarding on Euclidean distance and perimeter forwarding. The major consequence is a vicious cycle where these boundary nodes become overloaded, suffer power depletion at an early stage, and as a result cause the hole to grow further. An even stronger effect of this power depletion is that small neighboring holes merge into larger holes, resulting over time in large communication voids in the network.

In contrast, the fact that GLIDER uses intermediate landmarks to guide inter-tile routing means that hole-boundary nodes are of no particular importance and do not get overloaded. The cross-tile routing described in section VI-B allows packets to transit efficiently through each tile. Although the next tile's landmark is used to "pull" a packet through the current cell, once the packet enters the tile the target shifts to the landmark of the subsequent tile in the route. This avoids the undesirable effect of traffic convergence at the landmarks.

To test this, we randomly picked 45 source and destination pairs (in each case separated by more than 30 hops) within a single network. Figure 8 shows the "hot spots" when GLIDER

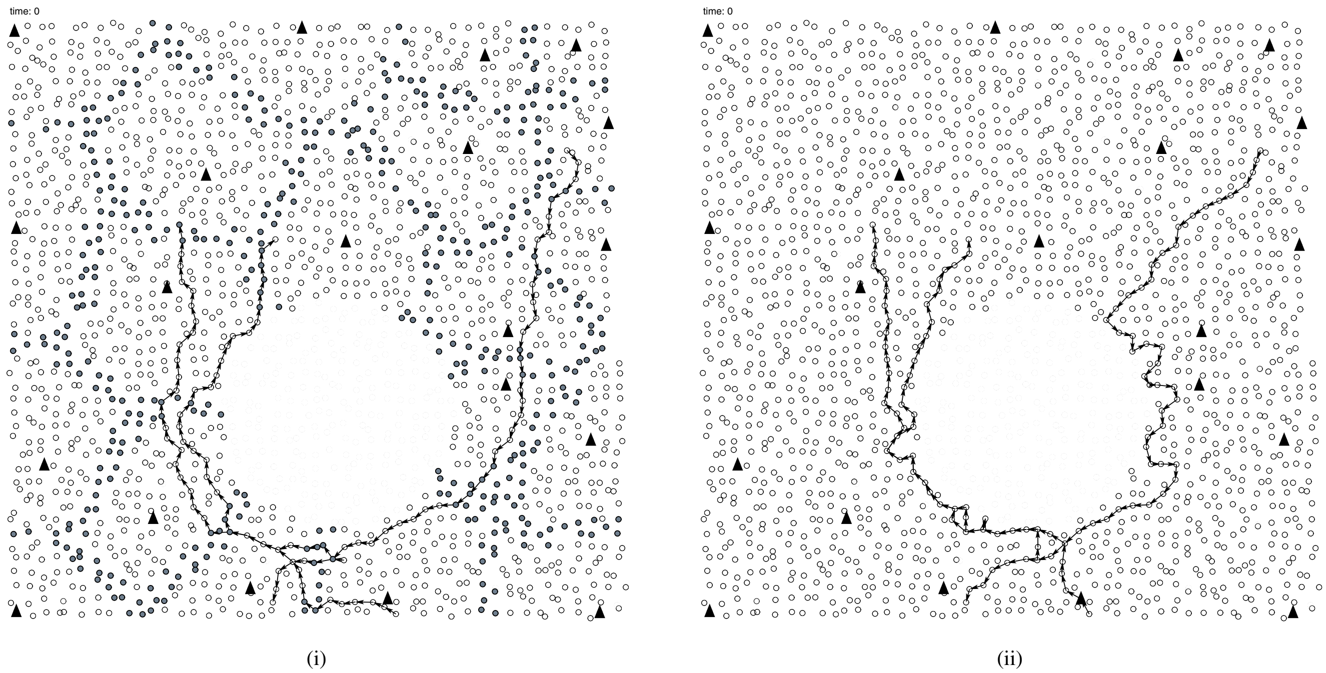


Fig. 7. The same network setup as in Figure 6 except for the standard deviation of the Gaussian random perturbation is 20% of the radio range (9m). Three source and destination pairs are shown in this figure. (i) Routes generated by GLIDER. (ii) Routes generated by GPSR.

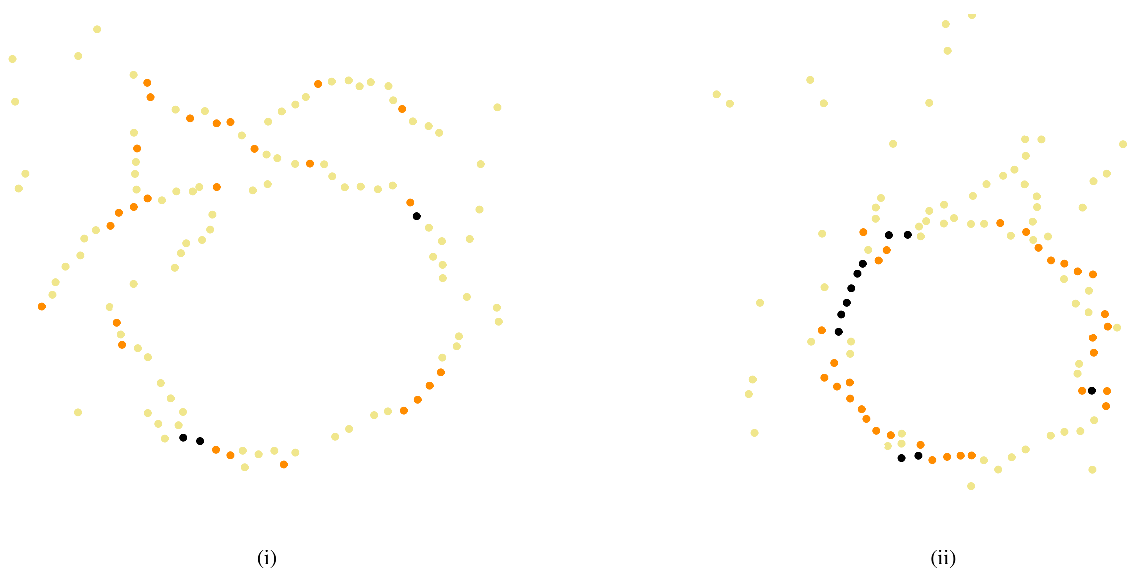


Fig. 8. Hot spots in the network with 45 pairs of randomly chosen source and destination; the darker the color, the heavier the traffic load: khaki ($6-8$ transit paths), orange ($9-11$ transit paths), black (≥ 12 transit paths) (i) traffic distribution map of GLIDER (ii) traffic distribution map of GPSR.

and GPSR are used, using different colors to indicate different levels of traffic load. Nodes lying on fewer than 6 routes are not colored. It is evident that the hole in the center creates a disturbance in traffic patterns. With GPSR, the effect is to create hot spots along the hole boundary. With GLIDER, traffic around the hole appears to be better spread out and more balanced.

Another disadvantage of boundary-hugging is that it tends to yield longer paths. In contrast, GLIDER uses shortest (in terms of the number of hops) paths for inter-tile routing. If the landmarks are placed close to topological features such as hole boundaries, GLIDER can circumnavigate obstacles comparatively efficiently. In the examples shown in Figure 7, GPSR generates a route of length 52 for the rightmost source to its destination, while GLIDER generates a route of length 41. In this figure, although the routes generated by GLIDER are longer in Euclidean distance, they are shorter in the graph-distance. On average, the lengths of routes generated by the two algorithms are comparable. For the 45 routes, the average path length generated by GPSR is 42.08. The average path length generated by GLIDER is 40.46. The major factor influencing path length in the case of GPSR is the geometric shape of the holes; in the case of GLIDER, it is the placement of the landmarks.

C. Discovery of routes under cases with difficult geometry

The landmark Voronoi complex succeeds in capturing the topology of the network and discovering routes even in situations that would be difficult for purely geometric approaches. In the scenario shown in Figure 2, two big rooms are connected by a long narrow corridor. Landmarks are selected randomly. The routing path that goes from one room to the other through the corridor is correctly discovered by our routing algorithm. We do not actually need landmark nodes be placed in the corridor itself. (Indeed, for randomly selected landmarks, the probability of finding a landmark in the corridor is comparatively small.) As long as the original network is connected, such connectivity is inherited by the combinatorial Delaunay triangulation.

IX. SUMMARY AND FUTURE WORK

In this paper we propose a topology-based naming and routing structure that uses only the link connectivity of the network. We do not use Euclidean coordinates—instead, we invent a more robust local landmark coordinate system within each tile, based on hop distances to nearby landmarks. We partition the network into tiles using the landmark Voronoi complex so that within each tile local greedy routing using our local coordinates can be expected to work well. We show that the Voronoi landmark-based routing protocol generates natural and load-balanced routing paths. The algorithms and protocols proposed in this paper work for sensor nodes in three dimensions as well—unlike other current geographic routing protocols (in fact, which underlying space the network nodes come from matters little).

Although we currently only exploit the path connectivity information stored in the landmark Voronoi complex for our routing scheme, we believe that the higher order connectivity information we compute will prove useful in more complex applications. An example may be loopy belief propagation and other probabilistic reasoning tasks that can benefit from a fuller understanding of the global topology of the sensor field.

It should be clear that this is only preliminary work on an approach to routing that leaves much to be explored. We still need to address important issues, such as the criteria and algorithms for landmark selection, potential multi-resolution LVC hierarchies for situations where a large number of landmarks is required, as well as methods for handling network dynamics (node addition and failure). Additional local coordinate systems also deserve to be explored, perhaps using partial or total information about the actual node positions or the communication quality between nodes.

ACKNOWLEDGEMENT

The authors wish to thank John Hershberger for helpful conversations. The authors also gratefully acknowledge the support of the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-00-1-0637, NSF grants CCR-0204486 and CNS-0435111, and DARPA grant #30759.

REFERENCES

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *3rd Int. Workshop on Discrete Algorithms and methods for mobile computing and communications (DialM '99)*, pages 48–55, 1999.

[2] G. Carlsson, A. Collins, L. Guibas, and A. Zomorodian. Persistence barcodes for shapes. In *Symposium on Geometry Processing*, 2004. to appear.

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

[4] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction, 2003.

[5] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *IEEE INFOCOM*, 2004.

[6] S. P. Fekete, A. Kroeller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSOR)*, pages 123–136, 2004.

[7] R. Fonesca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensornets, 2005.

[8] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[9] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 243–254, 2000.

[10] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an *ad hoc* sensor network. In *Proc. 2nd International Workshop on Information Processing in Sensor Networks (IPSN03)*, pages 333–348, Palo Alto, CA, April 2003. Springer.

[11] D. Niculescu and B. Nath. *Ad hoc* positioning system (APS). In *IEEE Global Telecommunications Conference (GlobeCom)*, pages 2926–2931, 2001.

[12] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on demand distance vector (AODV) routing, 1997.

[13] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108. ACM Press, 2003.

[14] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage. In *First International Workshop on Sensor Networks and Applications*, pages 78–87, 2002.

[15] E. M. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks, April 1999.

[16] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in *ad-hoc* networks of sensors. In *Proc. 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, pages 166–179, Rome, Italy, July 2001. ACM Press.

[17] A. Savvides and M. B. Strivastava. Distributed fine-grained localization in *ad-hoc* networks. submitted to IEEE Trans. on Mobile Computing.

[18] K. Seada, A. Helmy, and R. Govindan. On the effect of localization errors on geographic face routing in sensor networks. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 71–80. ACM Press, 2004.

[19] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 35–42. ACM Press, 1988.

[20] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier/Morgan-Kaufmann, 2004.