cse590 Information Processing in Sensor Networks Lecture 8

Location Free Routing in Sensor Networks

Lecturer: Jie Gao *Scribe: Radhika Bargavi*

1 Lecture Summary

In this class, we analyze how routing could be performed using global topology information. We discuss two Routing protocols, GLIDER and MAP which facilitate location free routing in sensor networks.

2 General Routing Methodology

We look for an efficient routing scheme for a dense sensor field with complex geometry having the following properties:

- Light-weight
- Localized routing
- Guaranteed delivery
- Load balancing
- Robust to dynamic change

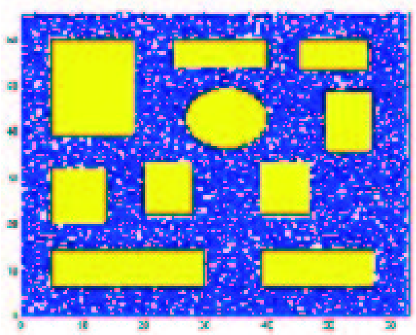

Figure 1: A dense sensor field (5204 nodes) with complex geometry

2.1 Two Level Routing Architecture

- **Top Level**: A compact abstraction of the global geometry/topology of the sensor field. E.g.: Hole in the middle of a sensor field.
- **Bottom Level**: A naming scheme with respect to the global topology that enables local gradient routing.

2.2 Actual Routing Mechanism

Check the compact abstract graph to get a global guidance on how to get around obstacles. However, the actual routing is local gradient descending.

Stable topological information is maintained proactively. The actual routing is local and reactive.

3 GLIDER

Gradient Landmark-Based Distributed Routing is a novel naming/addressing scheme and associated routing algorithm for wireless sensor networks.

3.1 Phases in GLIDER

- **Global Planning Phase**: Topology Discovery
- **Local Phase**: Routing as a sequence of hops using greedy method

Both the above phases 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 is that of the underlying sensor field.

3.2 GLIDER Glossary

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 tile must have trivial topology so that simple greedy routing will work well within the tile.

Our partition is defined by selecting a small set of well-dispersed nodes to be **landmarks**, and letting the tiles be the Voronoi cells for the landmarks, where the Voronoi cell of a landmark u is the set of nodes whose nearest landmark is u (in 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).

3.3 GLIDER implementation

Corresponding to the two phases in GLIDER, we introduce two protocols:

3.3.1 Naming Protocol

- Select landmarks to begin the topology discovery phase
- Construct the landmark Voronoi complex (LVC) in a distributed fashion
- Compute a routing table on the graph of the combinatorial Delaunay triangulation (CDT)
- Assign to each node its local landmark distance coordinates with respect to its reference landmarks

Information stored at a Node

✓ 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
✓ IDs of its neighbors

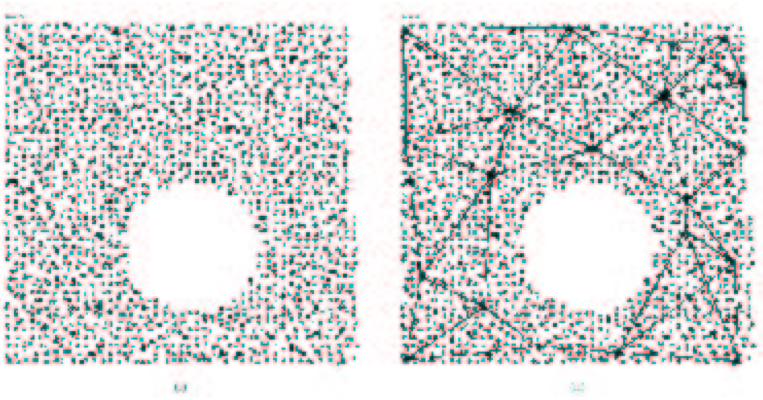

Figure 2: **(i)** Landmark Voronoi Complex (LVC) **(ii)** Combinatorial Delaunay triangulation (CDT)

3.3.2 Routing Protocol

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.
  - 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, the node v first selects whether the temporary destination landmark (TDL) 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}.
  - 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 which is closer than v to u_i in neighborhood distance.

Centered Landmark Distance Coordinate

Assume reference landmarks $L_0, \ldots, L_k$, and start tile T(p) = L_0. Let s = mean $(pL_0^2, \ldots, pL_k^2)$. Then the local virtual coordinates are c(p) = $(pL_0^2 - s, \ldots, pL_k^2 - s)$. This is a centered metric. Thus, the distance function is given by d(p,q) = $|c(p) - c(q)|^2$. By doing gradient descent on the distance function d(p,q), we can reach q using the greedy strategy.

Local Landmark Coordinates

We propose a virtual coordinate system which is easy to compute and is guaranteed to be free of local minima.

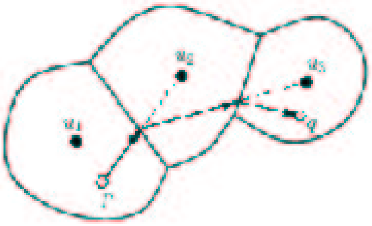

Figure 3: Routing across tiles

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

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

Centered coordinates $[C(p)]_i = -2p \cdot (u_i - \bar{u}) + w_i,$
where $\bar{u} = \frac{1}{k}\Sigma_j u_j$ and $w_i = |u_i|^2 - \frac{1}{k}\Sigma_j |u_j|^2$

The function p → C(p) is therefore an affine linear transformation.

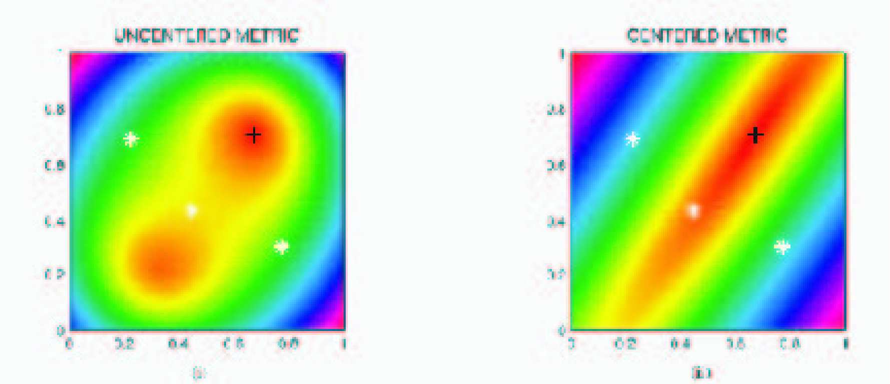

Figure 4: The distance function for landmark based greedy routing. There are 3 landmarks marked by snowflakes. Destination is marked by the + sign. **(i)** Uncentered coordinates case (note the local minima) **(ii)** Centered coordinates case

On an average, 5 or more neighbors are needed to ensure the success of greedy routing. The simulation results from a network of 2000 nodes distributed on a Gaussian(0, 0.5*radio range) perturbed grid are as follows:

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

Node density vs. Success Rate of greedy routing

4 MAP

Medial Axis Based Geometric Routing Protocol is similar to GLIDER in the way that MAP also takes the compact abstraction of the global topology of the sensor field. However, MAP does not rely on landmarks

Figure 5: Examples of routes generated by GLIDER. **L**:Narrow corridor connecting 2 rooms, **R**: Dense network with small and large holes

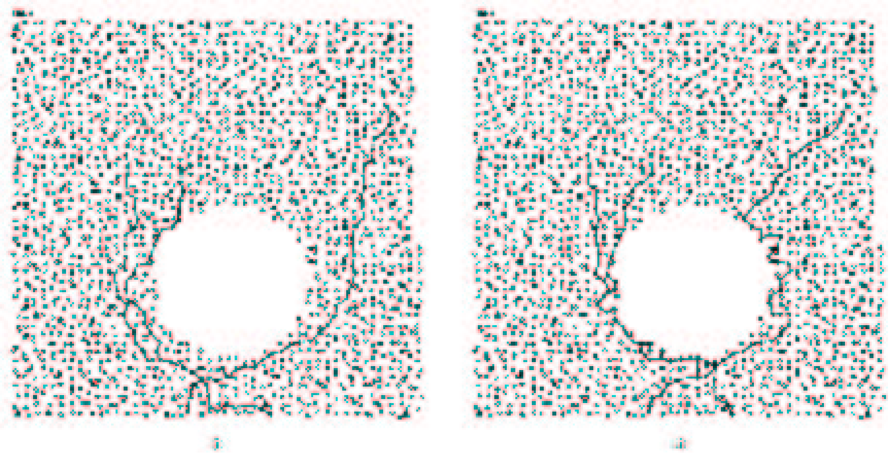

Figure 6: Simulations to find path lengths for **(i)** GLIDER 41 hops, **(ii)** GPSR 52 hops

but only on the shape of the sensor field. MAP depends only on the connectivity graph and does not require the communication network be a unit disk graph.

4.1 Protocols in MAP

MAP consists of two protocols:

- **MACP**: Medial Axis Construction Protocol constructs the medial axis and the corresponding naming scheme at the initialization stage of the network.
- **MARP**: Medial Axis based Routing Protocol routes packets by local gradient descent with the names of source and destination nodes.

4.2 MAP Glossary

Medial Axis is identified as the set of nodes with at least two closest points on boundaries. Given a bounded region R, the medial axis of its boundary δR is the collection of points with two or more closest points in

δR.

For each sensor w on the medial axis, we define a **chord** as the shortest path (tree) from w to one of its closest sensor node on the boundary. A chord is a line segment connecting a point on the medial axis and its closest points on δR. A point on the medial axis with 3 or more closest points on δR is called a **medial vertex**.

A sensors unique **name** includes the chord *x(p)y(p)* on which it stays, and a normalized distance *d(p)* to its corresponding medial axis sensor. E.g.: sensor p maybe named *(x(p), y(p), d(p))*.

Figure 7 shows an example of the medial axis of the boundary of a closed region R. The medial axis A has a cycle, which means that the region R has a punched hole.

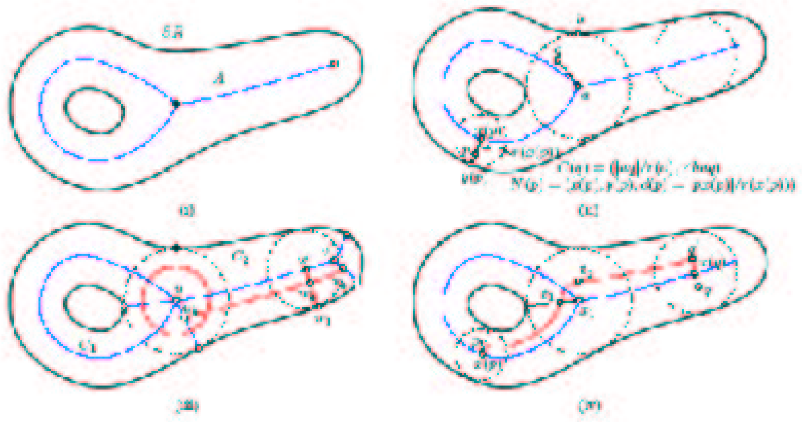

Figure 7: MAP in continuous case (i) The medial axis of δR and two medial vertices **(ii)** The naming scheme **(iii)** The road system on R. Two canonical cells *C1* and *C2* may share a common medial vertex but no common chord **(iv)** Routing from p to q.

4.3 Medial Axis Construction Protocol

MACP runs as follows:

- Detect boundaries of a sensor field
- Construct the medial access graph and broadcast it to every node in the network
- Name each node by only localized computation

Information stored at a Node

✓ The medial axis graph (MAG)
✓ Names of itself and one hop neighbors
✓ A bit to record if the node is on the medial axis
✓ The neighboring medial axis nodes

4.4 Medial Axis based Routing Protocol

MARP works as follows:

- In the global planning step, find the shortest path $S_A(x(p), x(q))$ in the medial axis graph A for the medial points of source p and destination q.

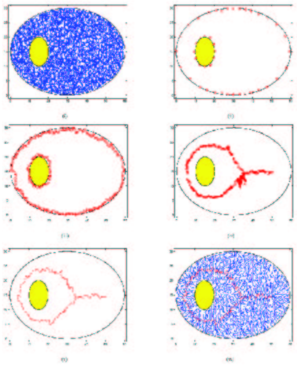

Figure 8: MAP in discrete case (i) A dense sensor network with 3000 nodes **(ii)** A small sample of hand-picked boundary nodes **(iii)** Discovery of more boundary nodes based on sample nodes **(iv)** Locally identified nodes on the medial axis **(v)** Medial axis after noisy nodes are eliminated **(vi)** Balanced shortest path trees rooted on the medial axis

- Route in parallel to $S_A(x(p), x(q))$ until a node with the same medial point as the destination q does is reached.
- Route along the shortest path trees rooted at that medial point to reach the destination q.

Manhattan Routing

– Routing in parallel to the medial axis
– Routing on chords

Routing between canonical pieces is done by first trying to match the medial axis point with that of the destination and route in parallel with the reference path, and then matching the point with the destination and routing along the chords. Refer to Fig. 7(iv).

4.5 Simulation Examples

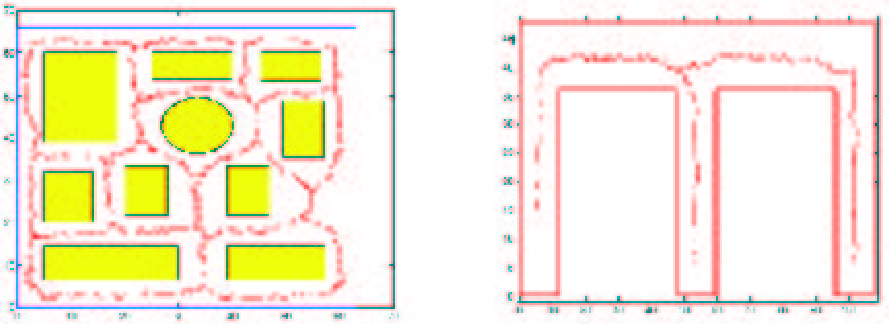

Figure 9: Examples of medial axis construction by MAP **L**: Scenario of University campus **R**: Scenario of airport terminals

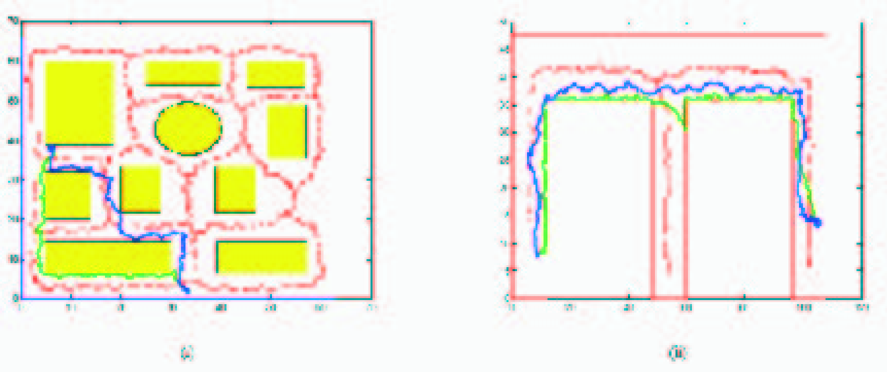
(a) (b)

Figure 10: Comparison of routing paths generated by **MAP** (blue) and **GPSR** (green)

5 References

- Qing Fang, Jie Gao, Leonidas Guibas, Vin de Silva, Li Zhang, **GLIDER: Gradient Landmark-Based Distributed Routing for Sensor Networks**, Proc. of the 24th Conference of the IEEE Communication Society (INFOCOM'05), March, 2005.
- J. Bruck, J. Gao, A. Jiang, **MAP: Medial Axis Based Geometric Routing in Sensor Networks**, to appear in the 11th Annual International Conference on Mobile Computing and Networking (MobiCom05), August, 2005.