Location-based Routing in Sensor Networks II

Jie Gao
Computer Science Department
Stony Brook University
Geographic routing in practice:


Virtual coordinates:

Overview of geographical routing

- Routing with geographical location information.
  - Greedy forwarding.
  - If stuck, do face routing on a planar sub-graph.
Overview of last lecture

• How to find a planar subgraph?
  – Use distributed construction: relative neighborhood graph, Gabriel graph, etc.
  – A planar subgraph that contains a short path: restricted Delaunay graph: short Delaunay edges.

• Big problem: how is the performance of geo-routing?
  – Can we always find a short path?
Bad news: Lower bound of localized routing

- Any deterministic or randomized localized routing algorithm takes a path of length $\Omega(k^2)$, if the optimal path has length $k$.

- The adversary decides where the chain $w_t$ is. Since we store no information on nodes, in the worst case we have to visit about $\Omega(k)$ chains and pay a cost of $\Omega(k^2)$.
Good news: greedy forwarding is optimal

- If greedy routing gets to the destination, then the path length is at most $O(k^2)$, if the optimal path has length $k$.

- $|uv|$ is at most $k$. On the greedy path, every other node is not visible, so they are of distance at least 1 away. By a packing lemma, there are at most $O(k^2)$ nodes inside a disk of radius $k$.

How is face routing? How is greedy + face routing?
Many variations of face routing in literature

A number of papers on various face routing:

- [Kuhn, et.al 02] Asymptotically optimal geometric mobile ad hoc routing.
- [Kuhn, et.al 03a] Worst-case optimal and average-case efficient geometric ad hoc routing.
- [Kim, et.al 05b] Geographic Routing Made Practical.
- [Kim, et.al 05a] On the Pitfalls of Geographic Face Routing.

Lesson: Do it carefully!!
Face transition

- In literature there are 4 ways of switching faces:
  1. Best intersection (AFR)
  2. First intersection (GPSR, GFG)
  3. Closest node other face routing (GOAFR+)
  4. Closest point other face routing
Face transition

- Simple first intersection may fail.

- Correct rule: at an intersection $p$, only change to a face that intersects $pt$ at $p$’s neighborhood.
Face transition

- Closest node other face routing fails in practice.

[Face Traversal based on GOAFR+]
walk 1 → 2: closest node to D = node 1
walk 2 → 3:
walk 3 → 4:
walk 4 → 5:
walk 5 → 6:
walk 6 → 7:
walk 7 → 1:
stop : since no closer than node 1
Face transition

- Best intersection face routing always makes progress towards the destination in a planar graph.
- The distance from the best intersection to the destination always decreases.
Performance of face routing

- What if we choose the wrong side?
Adaptive face routing

- Suppose the shortest path on the planar graph is bounded by $L$ hops.
- Bound the search area by an ellipsoid $\{x \mid |xs| + |xt| \leq L\}$ \(\Rightarrow\) never walk outside the ellipsoid.
- Follow one direction, if we hit the ellipsoid; turn back.
- If we find a better intersection $p$ of the face with line $st$, change to the face containing $pt$.
- In the worst case, visit every node inside the ellipsoid (about $O(L^2)$ by the bounded density property).
Adaptive face routing

- How to guess the upper bound L?
  - Start from a small value say |st|; if we fail to find a path, then we double L and re-run adaptive face routing.
  - By the time we succeed, L is at most twice the shortest path length k. The number of phases is $O(\log k)$.
- Total cost = $O(\sum_i (k/2^i)^2) = O(k^2)$. $\leftarrow$ asymptotically optimal.
A simple worst-case optimal routing alg

- It’s easy to get a worst-case $O(k^2)$ bound.

- Do adaptive restricted flooding.
  - Start with a small threshold $t$. Flood all the nodes within distance $t$ from the source.
  - If the destination is not reached, double the radius and retry.

- On a network with bounded density, the total cost is $O(k^2)$ if the shortest path has length $k$.

  Not quite efficient for most good cases.
Fall back to greedy

• When a node visits a node closer to the destination than that at which it enters the face routing mode, it returns to greedy mode.

• Other fall-back schemes are proposed. E.g., GOAFR+ considers falling back to greedy mode when considering a face change and when there are sufficient nodes closer to the destination than the local minimum.
Beyond point-to-point routing

- Multicast to a geographical region.
  - Use geographical forwarding to reach the destination region.
  - Restricted flooding inside the region.
- Routing on a curve.
  - Follow a parametric curve \( \langle x(t), y(t) \rangle \).

Figure 2: Forwarding on a curve
Geographical routing in practice
Revisit the assumptions of GPSR

- Nodes know their accurate locations.
- The network topology follows the unit disk graph model.

- These are 2 BIG assumptions.
- Localization is hard, both in theory and in practice.
- Unit disk graph model is simply not true in practice.
Sensor communication model

- Contour of probability of packet reception from a central node at two different transmit power settings.

Source: Ganesan, et.al Does not look like a disk to me.
Sensor communication model

• Each point represents a pair of nodes.

Source: Mark Paskin
Sensor communication model

- How in-bound link quality varies with distance.

Large variance even at short distances

Source: Mark Paskin
Sensor communication model

- Link quality varies with time.

Source: Mark Paskin
Sensor communication model

- Experiments show that
  - Irregular transmission range: stable long links exist, links between two close by nodes might not exist.
  - Links are asymmetric (A talks to B, B can’t talk to A).
  - Localization errors.

- This makes the planar graph construction fail.
Planar graph subtraction fails on irregular radio range

- Network is partitioned.
- Crossing links.

Edge AB is removed.

No crossing of line SD closer than point p.
Testing GPSR in a real testbed

- GPSR only succeeds on 68.2% directed node pairs.

A 50-node testbed at Intel Berkeley Lab
Planarization partitions the network

- Planar graph subtraction disconnects the network.

A 50-node testbed at Berkeley Soda Hall
A small fix on the asymmetric links

• The irregular radio range fails the planar graph construction.

• A small fix by using mutual witness:
  • The link AB is removed only if there is witness that is seen by both A and B.
A small fix on the asymmetric links

- Leaves more crossing links.
- Only improves the success rate of GPSR to 87.8%.
Cross Link Detection Protocol

• Try to do face routing on a non-planar network.

• Eliminate not-OK crossings and keep the graph connected.

• Each node probes each of its link to see if it’s crossed by other links.

• How to probe? Record the link to be probed in packet, do face routing and mark all crossings.
Cross Link Detection Protocol

- Start from D and do face routing.

Remove either AD or BC

Can’t remove BC

Can’t remove AD

Can’t remove either

Observation: a not-OK crossing is traversed twice, once in each direction.
Cross Link Detection Protocol

- A link is not removable, if it’s traversed twice.

- A crossing L and L’: remove the removable one. If none of them is removable, do nothing.

- Protocol: do the probing sequentially.

- For different probing sequences, one can get different graphs.

- Or, probe in a lazy fashion.
Multiple crossing links

- If a link is crossed by multiple other links, we probe it multiple times.

- Probing a pair of cross links may not find all the crossing, if they are obscured by other links.
Problems with CLDP

• How many probes? In what order?

• Can we probe the links concurrently?
  – Lock a link when it’s probed.

• Say we finish all the probes, and do face routing on the graph. Can we guarantee that the face routing always succeeds?
Summary on geographic routing

- Geographical routing is nice in terms of:
  - No flooding
  - No routing table maintenance
  - Scalable

- Face routing: Nice in theory, big mess in practice.
More thoughts

• We noticed that the trouble is due to face routing.

• Is greedy routing robust to localization noise?

• Can we ignore the real coordinates and use virtual coordinates for routing?
Approach I: Rubber band representation
Rubber band drawing of a graph

- All edges are rubber bands.
- Nail down some nodes $S$ in the plane, let the graph go.

- Theorem: the algorithm converges to a unique state – rubber band representation extending $S$.

Peterson graph with one pentagon nailed down.
The rubber band algorithm minimizes the total energy:

\[ E(x) = \sum_{i,j \in E} |x_i - x_j|^2 \]

Claim: \( E(x) \) is convex.

\[ E(x) = \sum_{i,j \in E} \sum_{k=1}^{d} (x_{ik} - x_{jk})^2. \]

When any \( x_i \) goes to infinity, \( E(x) \) goes to infinity. So we have a unique global minimum.

Peterson graph with one pentagon nailed down.
Rubber band drawing of a graph

- How does the rubber band representation look like?
- \( \partial E(x)/\partial x_i = 0 \).
- \( \sum_{j \in N(i)} (x_i - x_j) = 0 \).
- The rubber band connecting i and j pulls i with force \( x_j - x_i \). The total force acting on \( x_i \) is 0.
- The graph is at equilibrium.
Rubber band drawing of a graph

\[ \sum_{j \in N(i)} (x_i - x_j) = 0. \]

1. Every free node is at the center of gravity of its neighbors.

\[ x_i = \frac{1}{d_i} \sum_{j \in N(i)} x_j. \]

2. No reflex vertices.

Peterson graph with one pentagon nailed down.
More examples
Rubber band algorithm

- Recall the mass-spring model.
- First we assume nodes on the boundary know their location.
- Fix the nodes on the outer boundary.
- Iterative algorithm:
  - Every node moves to the center of gravity of its neighbors.

\[
x_i \leftarrow \frac{1}{d_i} \sum_{j \in N(i)} x_j.
\]

- Until no node moves more than distance \(\delta\).
A network with 3200 nodes

- Greedy routing success rate: 0.989, avg path length 16.8
Perimeter nodes are known (10 iterations)
Perimeter nodes are known (100 iterations)
Perimeter nodes are known (1000 iterations)

- Greedy routing success rate: 0.993, avg path length 17.1
Resiliency of the rubber band approach

- Greedy routing success rate: 0.981, avg path length 17.3
Resiliency of the rubber band approach

- Greedy routing success rate: 0.99, avg path length 17.1
How to fix perimeter nodes?

- Need nodes on the perimeter to “stretch” out the net.

- First assume we know nodes on the perimeter, but not the locations.
  1. Each perimeter sends hello messages.
  2. All the nodes record hop counts to each perimeter node.
  3. The hop count between every pair of perimeter node is broadcast to all perimeter nodes. (quite expensive)
  4. Embed perimeter nodes in the plane, say by any localization algorithm.

\[
\sum_{i,j \in \text{perimeter\_set}} (\text{measured\_dist}(i, j) - \text{dist}(i, j))^2
\]
Perimeter nodes

1. The embedding only gives relative positions: include 2 bootstrapping beacons in the embedding of perimeters.
   - Use the center of gravity as origin.
   - 1\textsuperscript{st} bootstrap node defines the positive x-axis.
   - 2\textsuperscript{nd} bootstrap node defines the positive y-axis.

2. Non-perimeter nodes actually have the distances to all perimeter nodes, and embed themselves.
   - Gives good initial positions for the rubber band algorithm.
How to find perimeter nodes?

- The bootstrapping nodes send hello messages to everyone.
- The node which is the farthest among all its 2-hop neighbors will identify itself as a perimeter node.
Success rate of greedy routing

- Success rate on virtual coordinates is comparable with true coordinates, when the sensors are dense and uniform.
Weird Shapes

(a)

(b)
Obstacles

- Success rate on virtual coordinates degrades when there are a lot of obstacles, but better than true coordinates.
Conclusions

- Geographical forwarding is quite robust to localization errors, or reasonable virtual coordinates.
- Geographical forwarding can easily scale to tens of thousands of nodes with acceptable overhead.
- For dense uniform sensor layout, we can eliminate the need for face routing altogether.
- Rubber band virtual coordinates respect the connectivity better than the true coordinates.
Next class

- Rubber band approach is kind of ad-hoc.
- In-depth study of “routing around holes”.
- Construction of virtual coordinates that respect to the topology of the field.
Presentations (2 virtual coordinates paper)

• James Newsome, Dawn Song, **GEM: Graph EMbedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information**, Proc. Sensys’03.


• A make-up lecture on Monday 10/9, 4-5:20pm??
Midterm project check

• Date
  – 10/10, 10/12?
  – Or, 10/17, 10/19?

• 20min talk per group: Present to the class what you plan to do and get feedback.
  – Problem definition.
  – Possible techniques.
  – Expected results.

• If you decide what topic you will work on, send me a note.