Routing and Localization in Sensor Networks

Anxiao (Andrew) Jiang

Computer Science Department
Texas A&M University

Joint work with: Jehoshua Bruck (Caltech) and Jie Gao (Stony Brook Univ.)
Multi-hop static ad hoc wireless network

- Shortest-path routing is expensive for large networks.
- Ad hoc wireless networks are closely related to the geometric environment where they are deployed.

Geometric information helps!
Location-based routing

Geographical Forwarding:

- Significantly simplifies the routing protocol, low routing overhead.
- Good for uniform and dense sensor deployment in a flat and regular region.
Location-based routing

- Geographical Forwarding may get stuck at local minima.

  Additional mechanisms to cope with dead-ends: Face Routing, Perimeter Routing, etc. [Bose, et.al 01][Karp, Kung 00][Kuhn et.al.03]
Face routing for guaranteed delivery

- Derive a planar subgraph of the network.
- Do face routing in the planar subgraph.

The original network
Face routing for guaranteed delivery

- Derive a planar subgraph of the network.
- Do face routing in the planar subgraph.

Remove some edges from the original network
Face routing for guaranteed delivery

- Derive a planar subgraph of the network.
- Do face routing in the planar subgraph.

Planar subgraph
Face routing for guaranteed delivery

- Derive a planar subgraph of the network.
- Do face routing in the planar subgraph.
Face routing for guaranteed delivery

- Derive a planar subgraph of the network.
- Do face routing in the planar subgraph.
Planar subgraph construction

Objective: Find a planar subgraph with “many” edges using localized computation.

Three constructions:
- Delaunay graph
- Gabriel graph
- Relative neighborhood graph
An assumption: unit disk graph model

Unit disk graph (UDG) model: All nodes have the same omnidirectional coverage range.
Location-based routing

- Localized, scalable
- Guaranteed delivery (but assuming UDG model)
- Location information is needed
Location-based routing

- Localized, scalable
- Guaranteed delivery (but assuming UDG model)
- Location information is needed
Geographical forwarding creates unbalanced load

- Poor performance in environments with complex geometry: Nodes on the boundaries are heavily loaded.

A campus with buildings.
Geographical forwarding creates unbalanced load

- Poor performance in environments with complex geometry: Nodes on the boundaries are heavily loaded.

Distribution of traffic load for 12000 random source and destinations.
Location-free routing: MAP

MAP:

Medial-Axis based geometric routing Protocol

J. Bruck, J. Gao and A. Jiang, MobiCom’05
MAP: medial-axis based geometric routing

- **Goal:** location free, delivery guaranteed, load balanced, light-weight, localized routing. 
  Not limited to UDG model.

- **Feature:** use geometric information characterized by medial axis, not node coordinates.
Routing holes

- Geographical forwarding uses the Euclidean coordinates as routing guidance, but routing is done on the connectivity graph.

  When there are holes in the sensor field, they mismatch.

- Thus the essential problem is, how to route around holes?

- We use the medial axis of the underlying geometric domain as a compact representation of global geometric/topological features.
MAP – Medial Axis based Naming/Routing

MAP --- Medial Axis based Naming/Routing Protocol
MAP – Medial Axis based Naming/Routing

MAP --- Medial Axis based Naming/Routing Protocol
Naming w.r.t. Medial Axis

A point \( p \) is named by the chord \( x(p)y(p) \) it stays on. \((x(p), y(p), d(p))\)

\( x(p) \) is a point on the medial axis.

\( y(p) \) is the closest point of \( x(p) \) on \( \partial R \).

\( d(p) \) is height, i.e., relative distance from \( x(p) \): \(|px(p)|/|x(p)y(p)|\).

Theorem: each point is given a unique name.
Routing between canonical pieces

Routing is done in 2 steps:

1. Check the medial axis graph, find a route connecting the corresponding points on the medial axis as guidance.

2. Realize the route by local gradient descending, in either the Cartesian coordinate system inside a canonical piece, or a polar coordinate system around a medial vertex.
Routing between canonical pieces

Routing is done in 2 steps:

1. Check the medial axis graph, find a route connecting the corresponding points on the medial axis as guidance.

2. Realize the route by local gradient descending, in either the Cartesian coordinate system inside a canonical piece, or a polar coordinate system around a medial vertex.
Challenges of MAP in discrete networks

- The hop count is only a rough approximation to the Euclidean distance.

- The exact medial axis is sensitive to noises.

- Low cost and distributed construction of a robust medial axis is desirable.

We use the same intuition as in the continuous case but keep these challenges in mind.
Detect boundaries of the sensor field.

- Find sample nodes on boundaries.

By manual identification, or automatic detection [Fekete’04, Funke’05]
Detect boundaries of the sensor field.

- Detect boundaries (a curve reconstruction problem).

Method: use local flooding to connect nearby boundary nodes, and include nodes on the shortest path between them as boundary nodes.
Detect boundaries of the sensor field.

- *Detect boundaries (a curve reconstruction problem).*

*Method: use local flooding to connect nearby boundary nodes, and include nodes on the shortest path between them as boundary nodes.*
MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Detect medial nodes (the sensors with 2 or more closest boundary nodes) by restricted flooding.*

The flooding is in fact a Voronoi partition of the network. So every node receives only one or a few flooded messages.

To suppress noise, for those nodes whose closest boundary nodes are on the same boundary and are very close to each other, we do not consider them to be medial nodes.
MAP in discrete networks --- naming

- Construct the medial axis graph.
  - Detect medial nodes (the sensors with 2 or more closest boundary nodes) by restricted flooding.
MAP in discrete networks --- naming

- Construct the medial axis graph.
  - Connect medial nodes into a graph and clean it up (remove very short branches).
MAP in discrete networks --- naming

- Construct the medial axis graph.
  - Connect medial nodes into a graph and clean it up (remove very short branches).

Medial axis graph: two vertices, two edges.
MAP in discrete networks --- naming

- Construct the medial axis graph.
  - Connect medial nodes into a graph and clean it up (remove very short branches).

Medial axis graph: two vertices, two edges.

Broadcast this simple graph to all sensors.
MAP in discrete networks --- naming

- Assign names to sensors for a discrete network:
  - Replace chords by (approximate) shortest path trees.

“Medial axis with dangling trees”
Assign names to sensors for a discrete network:

- Replace chords by (approximate) shortest path trees.
- Nodes are assigned names w.r.t. where it lies in its tree.
- Take advantage of the discreteness, assign names in a way to make it easy for insertion / deletion of nodes and edges.

All the computation is simple and local.
MAP in discrete networks --- routing

- **Medial Axis based Routing Protocol**
  - Mimic Manhattan routing.
  - Guaranteed delivery:
    If there is no better choice, route toward the medial axis.
  - Maintain balanced load:
    Try to route in parallel with the medial axis as much as possible, to avoid overloading nodes near the medial axis.
  - Building a small neighborhood routing table (e.g., a table for nodes within 3 hops) improves routing performance.
    Due to the discreteness of hop count distance.
Simulation Examples

- Outdoor sensor field: Campus

5735 nodes in the sensor network
Simulation Examples

Outdoor sensor field: Campus

Medial Axis
Simulation Examples

Outdoor sensor field: Campus

The simple medial axis graph: 18 nodes, 27 edges.
Simulation Examples

Outdoor sensor field: Campus

Routing path comparison:

Blue: MAP

Green: Location-based routing
Simulation Examples

Outdoor sensor field: Campus  ------ Load Balance Comparison

Location-based routing: Unbalanced Load

Nodes on boundaries are overwhelmed.
Simulation Examples

- Outdoor sensor field: Campus ------ Load Balance Comparison

MAP: Well Balanced Load
Simulation Examples

Outdoor sensor field: Campus       Load Balance Comparison

MAP: Location-based routing
Simulation Examples

Outdoor sensor field: Campus ---- Routing Distance Comparison

For the $i$-th path:

- Number of hops
  - MAP: $h_i$
  - GPSR: $H_i$

- Euclidean length
  - MAP: $l_i$
  - GPSR: $L_i$

Blue: Average ratio of hop numbers in a routing path
Red: Ratio of total hop numbers in all the routing paths

Blue: $\frac{1}{N} \sum_{i=1}^{N} h_i / H_i$
Red: $\frac{\sum_i h_i}{\sum_i H_i}$

Blue: $\frac{1}{N} \sum_{i=1}^{N} l_i / L_i$
Red: $\frac{\sum_i l_i}{\sum_i L_i}$
Simulation Examples

Indoor sensor field: Airport Terminals

5204 nodes in the sensor network
Simulation Examples

- Indoor sensor field: Airport Terminals

Medial Axis
Simulation Examples

- Indoor sensor field: Airport Terminals

The simple medial axis graph: 4 nodes, 3 edges.
Simulation Examples

Indoor sensor field: Airport Terminals

Routing path comparison:

Blue: MAP
Green: Location-based routing

source

destination
Simulation Examples

Indoor sensor field: Airport Terminals  ------ Load Balance Comparison

Location-based routing: Unbalanced Load

Nodes on boundaries are overwhelmed.
Simulation Examples

Indoor sensor field: Airport Terminals  ——— Load Balance Comparison

MAP: Well Balanced Load
Simulation Examples

Indoor sensor field: Airport Terminals  ------ Load Balance Comparison

MAP: Location-based routing
Simulation Examples

Indoor sensor field: Airport Terminals ---- Routing Distance Comparison

For the $i$-th path:

<table>
<thead>
<tr>
<th></th>
<th>Number of hops</th>
<th>Euclidean length</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP:</td>
<td>$h_i$</td>
<td>MAP: $\ell_i$</td>
</tr>
<tr>
<td>GPSR:</td>
<td>$H_i$</td>
<td>GPSR: $L_i$</td>
</tr>
</tbody>
</table>

Blue: $\frac{1}{N} \sum_{i=1}^{N} \frac{h_i}{H_i}$
Red: $\frac{\sum_{i} h_i}{\sum_{i} H_i}$

Blue: $\frac{1}{N} \sum_{i=1}^{N} \frac{\ell_i}{L_i}$
Red: $\frac{\sum_{i} \ell_i}{\sum_{i} L_i}$
Simulation Examples

So far, we have shown that MAP has better load balancing than geographical forwarding, and very similar routing distance (in terms of both hops and length) ……

What’s more, MAP is very robust to network models. It does not require the network to be a unit disk graph.
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: $\alpha = 0.8$

Maximum coverage range: 1.8

Minimum coverage range: 0.2
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: $\alpha = 0.8$

Medial Axis (for campus):
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: \( \alpha = 0.8 \)

Medial Axis (for campus):

Although the network is very different from unit disk graph, the construction of medial axis is very robust.
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: $\alpha = 0.8$

Compare MAP Load: both well balanced

$\alpha = 0.8$  $\alpha = 0$ (UDG)
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: \( \alpha = 0.8 \)

Compare Load Balance:

MAP: Well Balanced
\( \alpha = 0.8 \)

Location-based routing: Unbalanced
Simulation Examples

MAP on quasi unit disk graphs: Routing Distance

Number of hops
For the $i$-th path: Quasi-UDG: $h_i$  UDG: $H_i$

Euclidean length
Quasi-UDG: $\ell_i$  UDG: $L_i$

![Campus](image1)

![Airport Terminal](image2)

$x$-coordinate: $\alpha$

Blue: $\frac{\sum_i h_i}{\sum_i H_i}$  Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$

Blue: $\frac{\sum_i h_i}{\sum_i H_i}$  Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$
Summary of MAP

1. Medial axis captures the shape of the sensor field.
2. It is compact.
3. No location-information is needed.
4. No unit disk graph assumption.
5. Good load balancing.
Part II :

Localization of Sensor Networks
Why is location information important?

- Sensor networks are data-centric.
  Example: temperature

- The location information can help routing.
  Example: geographical routing
Sensor network model

Sensor network model: unit-disk graph (UDG).

Definition: There is an edge between two nodes if and only if their distance is at most 1.
What is localization?

sensors

a b c d e f g h
What is localization?

Sensors can learn their positions by using GPS.

But GPS is expensive and sometimes unavailable (e.g., indoors).

So we resort to alternative methods …
What is localization?

sensor network (UDG)

connectivity information
What is localization?

Localization (also known as embedding)

sensor network (UDG)

connectivity information
Localization by local angle information

- Past research: localization by distance information.

- We study: localization by local angle information.

  Local angle: the angle between two adjacent edges.
Localization by local angle information
Localization by local angle information

What we want: a valid embedding.

Definition:

*Input*: A graph, and the values of local angles.

*Output*: An embedding of the graph that satisfies

- Two nodes have an edge iff their distance is at most 1
- The local angles are as specified.
Overview of results

unit-disk graph, local angles → embedding

NP
Overview of results

unit-disk graph, embedding \( \xrightarrow{\text{NP}} \) planar spanner
local angles \( \xrightarrow{\text{P}} \) (for location-based routing)
Overview of results

unit-disk graph, local angles $\xrightarrow{NP}$ embedding $\xrightarrow{P}$ planar spanner (for location-based routing)
NP hardness of embedding

**Input:** A graph, and the values of local angles.

**Output:** A valid embedding.
NP hardness of embedding

Basic idea: a reduction from the 3SAT problem.

A 3SAT instance:

\((\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor x_3)\)
Formulate the 3SAT problem as a graph

3SAT instance: \((\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor x_3)\)

3SAT graph:

Realize the 3SAT graph with a unit-disk graph.
Formulate the 3SAT problem as a graph

3SAT instance: \( (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor x_3) \)

3SAT graph:
0/1 block --- true realization by UDG

Case 1: \( x_1 = 1 \) \( \overline{x}_1 = 0 \)

Case 2: \( x_1 = 0 \) \( \overline{x}_1 = 1 \)
Formulate the 3SAT problem as a graph

3SAT instance: \((\overline{x_1} \lor x_2 \lor x_3) \land (x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor x_3)\)

3SAT graph:
Realize a 3SAT clause by a UDG

3SAT clause: \((\overline{x_1} \lor x_2 \lor \overline{x_3})\)

\(v_1, \ v_2, \ v_3: \ \leq \frac{1}{2}l \quad \text{or} \quad \geq \frac{2}{3}l\)
NP hardness of embedding

Embedding: NP hard.

Another conclusion (details skipped):
\[ \sqrt{2} \] approximate embedding is also NP hard.
Practical embedding using local angles

Embedding using local angles works well in practice.
Practical embedding using local angles

Embedding using local angles works well in practice.

True Network (600 nodes)  Embedding
Practical embedding using local angles

Embedding using local angles works well in practice.

True Network (600 nodes)  Embedding