Wireless Sensor Networks

7. Geometric Routing

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg
Version 30.05.2016
Literature - Surveys


Geometric Routing

- Routing target:
  - geometric position

- Idea
  - send message to the neighbor closest to the target node (greedy strategy)

- Advantages:
  - only local decisions
  - no routing tables
  - scalable
### Prerequisites

- Each node knows its position (e.g. GPS)
- Positions of neighbors are known (beacon messages)
- Target position is known (location service)
Greedy forwarding and recovery

- With position information
  - one can forward a message in the "right" direction (greedy forwarding)

no routing tables, no flooding!
First Approaches

- Routing in packet radio networks
- Greedy strategies:
  - MFR: Most Forwarding within Radius  [Takagi, Kleinrock 1984]
  - NFP: Nearest with Forwarding Progress  [Hou, Li 1986]
Greedy forwarding and recovery

- Greedy forwarding is stopped by barriers
  - (local minima)
- Recovery strategy:
  - Traverse the border of a barrier until a forwarding progress is possible (right-hand rule)
  - Routing time depends on the size of barriers
Position Based Routing

- Combination of greedy routing and recovery strategy
- Recovery from local minima (right hand rule)
  - Example: GPSR [Karp, Kung 2000]

Greedy forwarding and recovery

- Right-hand rule needs planar topology
  - otherwise endless recovery cycles can occur
- Therefore the graph needs to be made planar
  - erase crossing edges
- Problem
  - needs communication between nodes
  - must be done careful in order to prevent graph from becoming disconnected
Problems of Recovery

- Recovery strategy can produce large detours
- Solutions
  - Follow recovery strategy until the situation has absolutely improved
    - e.g. until the target is closer
  - Follow a thread
    - Face Routing strategy, GOAFR
GOAFR: Adaptive Face Routing

- **Adaptive Face Routing**
  - Faces are traversed completely while the search area is restricted by a bounding ellipse
  - Recovery strategy + greedy forwarding

**Figure 10:** Adaptive Face Routing (AFR) [52]. Faces are traversed completely while the search area is restricted by a bounding ellipse.

3.4 Recovery beyond Planarization

Recovery from local minima does not necessarily require creating a planar subgraph prior to routing. One approach is to identify possible local minima beforehand, such that a path around a void region is already established when a packet in greedy mode arrives at a dead-end node. For this purpose Fang et al. [16] propose the Bound...
Planarization

- Construction of planar subgraph
  - Gabriel graphs
    - edges where closed disc of which line segment \((u,v)\) is a diameter contains no other elements of \(S\)
  - Relative Neighborhood Graph
    - edges connecting two points whenever there does not exist a third point that is closer to both
  - Delaunay Triangulation
    - only triangles such that no point is inside the circumcircle
Adaptive Face Routing

- Spanning ratio/stretch factor
  - $\max\{\frac{\text{shortest path}(u,v)}{\text{geometric distance}(u,v)}\}$
- Gabriel graphs $\Theta(\sqrt{n})$
- Relative Neighborhood Graph $\Theta(n)$
- Delaunay Triangulation $\frac{4\pi}{3\sqrt{3}}$
  - but possibly long edges
  - because the convex hull is always a sub-graph of the DT
- A lot of better techniques studied in literature

- The Restricted Delaunay Graph (RDG) [29] is obtained by locally constructing a sub-graph of the DT but possibly long edges.
- The Partial Delaunay Triangulation (PDT) [60] has been proposed in two variants.
- To alleviate this problem, local construction schemes for the Delaunay triangulation were considered, which can be constructed locally [29, 59, 60]. They are considered. With 1-hop information, an edge is removed. If these nodes are located either only left or only right of the circle, the maximum Delaunay circle among these nodes is empty (see Figure 8). In contrast to the GG or RNG, this criterion cannot be checked with 1-hop information, whereas the RNG, i.e., the detours induced by these subgraph constructions are not bounded by a constant.
- In terms of hop count, both GG and RNG have unbounded stretch factors, as it guarantees a constant overhead for any path in the subgraph.
- As planarization removes crossing edges it may induce detours due to missing edges. Therefore, planar subgraph constructions are desired that approximate the spanning ratio, which is known to have a constant spanning ratio. The Delaunay triangulation of a given point set contains all triangles whose circumcircle is empty. With 2-hop information, covers the Delaunay circle (i.e. it is able to reach every other node within this circle).
- To study the relative neighborhood graph, we consider that the so-called adaptive face routing is based on the Gabriel graph (left), RNG (middle) and Delaunay triangulation (right) for the Gabriel graph and RNG, however, the spanning ratio is unbound. Bose et al. have proven length stretch factors of $Q_n$ for the Gabriel graph and $Q_{n^2}$ for the RNG, i.e. the detours induced by these subgraph constructions are not bounded by a constant. In terms of hop count, both GG and RNG have unbounded stretch factors. For Gabriel graph and RNG, however, the spanning ratio is unbound.

- $d$ = length of shortest path
- Time = #hops, traffic = #messages

Time: $\Omega(d^2)$
Lower Bound for Greedy Routing


Time: $\Omega(d^2)$

d = length of shortest path

time = #hops, traffic = #messages
nodes exchange beacon messages
⇒ node $v$ knows positions of ist neighbors

transmission radius
(Unit Disk Graph)

Rührup et al. Online Multi-Path Routing in a Maze, ISAAC 2006
A Virtual Cell Structure

Each node **classifies** the cells in its transmission range.
Routing based on the Cell Structure

- Routing based on the cell structure uses cell paths
- cell path
  - = sequence of orthogonally neighboring cells

- Paths
  - in the unit disk graph and cell paths are equivalent up to a constant factor

- no planarization strategy needed
  - required for recovery using the right-hand rule
Routing based on the Cell Structure

**virtual** forwarding using cells

**physical** forwarding from $v$ to $w$, if visibility range is exceeded

- node cell
- link cell
- barrier cell
Performance Measures

- **competitive ratio:**
  - solution of the algorithm
  - optimal offline solution

- **competitive time ratio of a routing algorithm**
  - $h = \text{length of shortest barrier-free path}$
  - algorithm needs $T$ rounds to deliver a message

$$R_t := \frac{T}{h}$$

[Diagram showing single-path $T$]
Comparative Ratios

- optimal (offline) solution for traffic:
  - $h$ messages (length of shortest path)

Unfair, because
- offline algorithm knows the barriers
- but every online algorithm has to pay exploration costs

- exploration costs
  - sum of perimeters of all barriers ($p$)

- comparative traffic ratio

$$R_{Tr} := \frac{M}{h + p}$$

- $M = \# \text{ messages used}$
- $h = \text{ length of shortest path}$
- $p = \text{ sum of perimeters}$
Comparative Ratios

- measure for time efficiency:  
  - competitive time ratio
  \[ R_t := \frac{T}{h} \]

- measure for traffic efficiency:  
  - comparative traffic ratio
  \[ R_{Tr} := \frac{M}{h + p} \]

- Combined comparative ratio  
  - time efficiency and traffic efficiency
  \[ R_c := \max\{R_t, R_{Tr}\} \]
Single Path Strategy

- no parallelism
  - traffic-efficient (time = traffic)
  - example: GuideLine/Recovery
- follow a guide line connecting source and target
- traverse all barriers intersecting the guide line
- Time and Traffic: $O(h + p)$
Multi-path Strategy

- speed-up by parallel exploration
  - increasing traffic
  - example: Expanding Ring Search
- start flooding with restricted search depth
- if target is not in reach then
  - repeat with double search depth

- Time $O(h)$
- Traffic $O(h^2)$
Algorithms under Comparative Measures

<table>
<thead>
<tr>
<th>Algorithm/Recovery</th>
<th>Time</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>GuideLine/Recovery (single-path)</td>
<td>$\mathcal{O}(h + p)$</td>
<td></td>
</tr>
<tr>
<td>Expanding Ring Search (multi-path)</td>
<td>$\mathcal{O}(h)$</td>
<td>$\mathcal{O}(h^2)$</td>
</tr>
</tbody>
</table>

Is that good?

\[
R_t := \frac{T}{h} \\
R_{Tr} := \frac{M}{h + p}
\]

It depends ... on the scenario:

<table>
<thead>
<tr>
<th>Algorithm/Recovery</th>
<th>Scenario</th>
<th>Time Ratio</th>
<th>Traffic Ratio</th>
<th>Combined Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>GuideLine/Recovery (single-path)</td>
<td>maze</td>
<td>$\mathcal{O}(h)$</td>
<td>$\mathcal{O}(1)$</td>
<td>$\mathcal{O}(h)$</td>
</tr>
<tr>
<td></td>
<td>$p = h^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expanding Ring Search (multi-path)</td>
<td>open space</td>
<td>$\mathcal{O}(1)$</td>
<td>$\mathcal{O}(h)$</td>
<td>$\mathcal{O}(h)$</td>
</tr>
<tr>
<td></td>
<td>$p &lt; h$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Alternating Algorithm

- uses a combination of both strategies:
- 1. \( i = 1 \)
- 2. \( d = 2^i \)
- 3. start GuideLine/Recovery with time-to-live = \( d^{3/2} \)
- 4. if the target is not reached then
  
  start Flooding with time-to-live = \( d \)
- 5. if the target is not reached then
  
  \( i = i+1 \)
  
  goto line 2

- Combined comparative ratio: \( R_c = \mathcal{O}(\sqrt{h}) \)
The JITE Algorithmus

Rührup et al. Online Multi-Path Routing in a Maze, ISAAC 2006

- Complex algorithm
- Message efficient parallel BFS (breadth first search)
  - using Continuous Ring Search
- Just-In-Time Exploration (JITE)
  - construction of search path instead of flooding
- Search paths surround barriers
- Slow Search
  - slow BFS on a sparse grid
- Fast Exploration
  - Construction of the sparse grid near to the shoreline
- Slow Search visits only explored paths

- Fast Exploration is started in the vicinity of the BFS-shoreline

- Exploration must be terminated before a frame is reached by the BFS-shoreline
Performance of Geometric Routing Algorithms

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Time</th>
<th>Traffic</th>
<th>Comb. Comp. Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. Ring Search [9, 18]</td>
<td>$O(d)$</td>
<td>$O(d^2)$</td>
<td>$O(d)$</td>
</tr>
<tr>
<td>Lucas’ Algorithm [13]</td>
<td>$O(d + p)$</td>
<td>$O(d + p)$</td>
<td>$O(d)$</td>
</tr>
<tr>
<td>Alternating Strategy [20]</td>
<td>$O(d^{3/2})$</td>
<td>$O(\min{d^2, d^{3/2} + p})$</td>
<td>$O(\sqrt{d})$</td>
</tr>
<tr>
<td>Selective Flooding [21]</td>
<td>$d \cdot 2^{O(\sqrt{\log \log d})}$</td>
<td>$O(d) + p d^{O(\sqrt{\log \log d})}$</td>
<td>$d^{O(\sqrt{\log \log d})}$</td>
</tr>
<tr>
<td>JITE (this paper)</td>
<td>$O(d)$</td>
<td>$O((d + p) \log^2 d)$</td>
<td>$O(\log^2 d)$</td>
</tr>
<tr>
<td>Online Lower Bound (cf. [3])</td>
<td>$\Omega(d)$</td>
<td>$\Omega(d + p)$</td>
<td>$\Omega(1)$</td>
</tr>
</tbody>
</table>

Rührup et al. Online Multi-Path Routing in a Maze, ISAAC 2006
Beacon-Less Geometric Routing


Beaconless Routing

- **Givens**
  - Each node knows its position
  - A node knows the position of the routing target
  - No beacons
  - The neighborhood is unknown
  - Nodes listen to messages
  - Sparse routing information in packets

- **The Idea**
  - A packet carries the source and target coordinates
  - Only good located sensor answers
Beaconless Routing
The Roles

- **Forwarder**
  - node currently holding the packet

- **Forwarding Area**
  - nodes in this area are allowed to accept the packets

- **Candidates**
  - nodes in the forwarding area
  - most suitable candidate chosen by contention

- **Timer**
  - each candidate has a time based on a delay function
  - The delay function has as parameters the coordinate of the forwarder the target and the own position

---

Forwarding Area

- target and the own position
- the coordinate of the forwarder
- each candidate has a time based on a contention
- most suitable candidate chosen by
  node currently holding the packet

---

Forwarder

- node currently holding the packet
- nodes in this area are allowed to accept the packets
- nodes in the forwarding area
- most suitable candidate chosen by contention
- each candidate has a time based on a delay function
- The delay function has as parameters the coordinate of the forwarder the target and the own position

---

H. Kalosha et al. Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks InfoCom 2008
Beaconless Routing
Problem: Recovery Strategy

- Greedy Routing works perfectly
- But recovery strategy is problematic
  - How to construct local planar subgraphs on the fly
  - How to determine the next edge of a planar subgraph traversal
- Rules
  - no beacons allowed to solve this problem
  - but interaction with the neighborhood
Possible Recovery Strategies

- **BLR Backup Mode**
  - Literature

- **Algorithm**
  - Forwarder broadcast to all neighboring nodes
  - All neighbors reply
  - Construct a local planar subgraph (Gabriel Graph)
  - Forward using right-hand-rule

- BLR guarantees delivery
  - but needs reaction of all neighbors (pseudo-beacons)
Possible Recovery Strategies

- **NB-FACE**
  - Literature
    - M. Narasawa, M. Ono, and H. Higaki, “NB-FACE: No-beacon face ad-hoc routing protocol for reduction of location acquisition overhead,” in 7th Int. Conf. on Mobile Data Management (MDM’06), 2006, p. 102.

- **Algorithm**
  - Delay function depends on the angle between forwarder candidate and previous hop,
    - such that the first candidate in clockwise or counterclockwise order responds first.
  - If this node is not a neighbor of the Gabriel graph, then other nodes **protest**

- **NB-Faces also guarantees delivery**
  - this strategy was improved by Kalosha et al. in order to decrease the number of messages
Location Service

- How to inform all nodes about the position of the destination node(s)

**Categories**

- Flooding-based location dissemination
  - fastest and simplest way
  - yet many messages
- Quorum-based and home-zone-based strategies
  - reduces communication overhead
- Movement-based location dissemination
  - location information is spread only locally
  - table of location and time stamps is exchanged when two nodes come close to each other
  - only applicable to mobile networks
Quorum based Location Services

- Location information at group of nodes
- Nodes need to be contacted to obtain information
- E.g. consider grid (Stojmenovic, TR 99)
  - Destination information information is stored on a row
  - Node needs to ask all nodes in a column to receive this information
  - reduces traffic by a factor of $O(n^{1/2})$

- Grid Location Service (Li et al. MobiCom 00)
  - location servers distributed by a hierarchical subdivision of the plane
Each node has a home-zone
- in this home zone (possibly far away)
- another nodes is responsible for relaying position information

Geographic Hash Tables (Ratnasami et al. 02)
- Node and location are key-value pair
- key is assigned to a location by a hash function
- In this location the home zone router is responsible for storing this information
- Each node updates his information at the home zone router
- Nodes looking for a node contact home zone router
Summary

- Geometric Routing is a scalable alternative with only local information
- Recovery strategies
  - are necessary since barriers might occur
- Planarization
  - underlying communication graph should be planar
  - erase edges or use cell structure
- Beacon- and baconless Routing
- Location Service is necessary
- Real-world Solutions
  - Flooding
  - Alternating algorithm
  - Greedy with right-hand recovery
  - Greedy with flooding recovery
Wireless Sensor Networks
7. Geometric Routing

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg
Version 30.05.2016