EECS 369

Introduction to Wireless Sensor Network

(Winter 2011)

Peter Scheuermann and Goce Trajcevski

Dept. of EECS

Northwestern University
Geography-Aware Routing Protocols
Recall: Taxonomy of Routing Protocols for Wireless Sensor Networks

1. DATA CENTRIC PROTOCOLS
   e.g., Flooding, Gossiping, SPIN, Directed Diffusion,…

2. HIERARCHICAL PROTOCOLS
   e.g., LEACH, TEEN,…

3. LOCATION BASED (GEOGRAPHIC) PROTOCOLS
   1. GPSR
   2. TBF
      (see the corresponding papers)
   PLUS “Potpourri”…
Geographical Routing - Basics

Next Hop Selection
Given a DESTINATION, the node that is holding the message selects the next hop according to

1) Its own position
2) The position of the destination node
3) The position of its neighbors (nodes in the Knowledge Range)

DIFFERENT FORWARDING RULES ARE POSSIBLE!
Key Assumptions:

- Nodes (routers) know their location
  (OUCH!) GPS, beacon, tri/multi-lateration…
- (Roughly) Planar Topologies
- Maybe: Registration/Lookup service mapping nodes to location
  - Sources can determine the addresses of their destinations and encode them as part of the packet(s).
- Queries use the same “address-book”
  - (Implicit: Unit-disk graph model of communication range…)
Node X receives a packet, for which the destination is D

Of all the X's neighbors, Y is closest to D

=> “greedy” forwarding (decreasing the total distance)
Q: What if the neighborhood changes (e.g., nodes deplete their energy; new nodes enter the region; …)?

- Periodically, each node transmits a beacon to the common, broadcast MAC address, containing (ID, location). Hence, the neighbors can update their data…
- If the time during which a beacon has not been received from a given neighbor exceeds a pre-defined time-out interval, assume failure and delete it from neighborhood-table.
  - Two four-Bytes fields (float) for each of X and Y coordinates…

NOTE: this is pro-active…

To save on communication for beaconing, location info can be piggy-backed on the data packets

- All?
- Which ones?
BPSR – Problem(s) with the “greedy”...

For the given network, assume that X receives a packet to be forwarded to the node D.

IF A & B are the only ones in Its communication range, since X Is closer to D than both of them \( \Rightarrow \)the “pure” GPSR would NOT send the packet !!!

Hence, one type of problems are due to the, so called, VOID regions.
GPSR – Problem(s) with the “greedy”…

Solution to *voids*:
- Travel around the perimeter of the void, using as “road-segments” the edges between the nodes (view communication graph as a node)
- Eventually/hopefully, get closer to the desired destination… (e.g., X→B→E→G→D)

OK, so this is kind’a graph-theoretic…
Ergo, it brings another problem: how Are edges that are intersecting to be treated (are they having an actual vertex)

Solution: Enforce the *PLANARITY*…
GPSR – Problem(s) with the “greedy”...

- Desideratum: reduce the number of “active” neighbors, while preserving the connectivity of the network as a whole.
  - This should be done in a manner to ensure min. amount of “links” to be traveled for whatever purpose needed…

- Two basic geometric techniques used for making a given graph planar, while ensuring that all the nodes that the connectivity is the same, with respect to the initial connectivity under the unit-disk model:
  - Relative Neighborhood Graph (RNG)
  - Gabriel Graph (GG)
    - (other methods, e.g., Yao graphs…)
GPSR – Planarization of Graphs:

**RNG:**
An edge exists between $u$ and $v$, if their distance is less than the $\max[(u,w),(v,w)]$ for any other such vertex $w$

*e.g., no “witnesses” in the luna*

**GG:**
An edge exists between $u$ and $v$, if no other vertex is inside the circle whose diameter is $uv$

*e.g., no “witnesses” inside the circle*

Clearly, GG more restrictive than RNG!!
Quantitative Observations…

Radio range = 250 meters
Back to the USSR…

OK, so given an initial network, assume that we are done with RNG-ization or GG-ization…

The typical packet can either:
forward greedily;
or
forward around perimeter…

For the purpose of forwarding around the perimeter, the GPSR packet header has the following fields:

- $D$ - destination location;
- $L_p$ - Location in which the packet entered the “perimeter” mode;
- $L_f$ - Location on $xV$ in which the packet entered current face (TBE);
- $e_0$ - first edge traversed on the current edge;
- $M$ - packet mode (G/P)
So, just what is “walk around perimeter”??
AKA Face Routing…

- Keep left hand on the wall, walk until hit the straight line connecting source to destination.
- Then switch to the next face.

Und so weiter, und so weiter…

…proceed “recursively”
GPSR

Face: planar region bounded by the edges in a given graph (can be “open”)

When void encountered:
- “draw” the line XD;
- Pick the face at X intersected by XD;
- Select the edge on that face -> LHS;
- Traverse the edges on the boundary of that face -> RHS;
- At any point, if non-void (i.e., greedy-possible), do greedy;

NOTES:
1. Cycles can be detected (recall the header data)
2. Cycles can only happen when X and D are NOT in one connected component...
Some Issues of GPSR...

- What if mobility is part of the game?
  - MAC failure feedback...
- Promiscuous use of network interface
  - Disable MAC address filtering (reason: every packet carries location data...)
- How realistic is the assumption about symmetric links (in turn, how good is the RNG/GG-ization of the connectivity graph)?
- Planarity of the graph?
  - Nodes move (in/out), deplete batteries => batch or incremental updates?
Some Issues with the GPSR…

Alternative “progress” measures

Issue++: Greed is not a good habit…
(face routing, although more “expensive” ensures that one cannot end up in a dead-end…)

Send packets to the neighbor closest to the destination
Trajectory-Based Forwarding (TBF)

- A paradigm/general-recipe, rather than an actual implementation…
- Target = minimize the overheads which arise in:
  - Discovery
  - Construction of the route(s)
  - Scalability
    - Routing structure maintenance/update; space-time-flooding…
- Crux:
  - Instead of specifying
    - Destination, OR
    - Event/Region, OR
    - …
  - Specify the TRAJECTORY that the packets should follow…
TBF – Basic Idea...

Ideally:

- discrete paths
- overhead $\sim$ path length
- mobility $\rightarrow$ updates

Possibility:

- continuous paths
- fixed overhead
- no maintenance
How TBF Forwards…

Needs to transmit the parameters of the curve representing the trajectory, e.g., $Ax^2 + Bx + C$ (in case the desired trajectory should resemble a parabola)

Problem: as the “nodes advance” (grain-of-salt-here), how do they know which value of x (or y) corresponds to them…

Hence, a better choice may be to represent the curve in a parametric form $X = f_x(t), Y = f_y(t)$…

- **node $N_0$:**
- **receives** $X(t), Y(t)$ - encoded curve
- **chooses a next hop**
  - closest to trajectory ($N_2$)
  - maximum advancement ($N_4$)
  - most battery left
Possible Problems for TBF…

Sparse Networks…

true space

warped space

landmark node
forwarding node
ideal trajectory
achieved trajectory
Extended Benefits of TBF…

TBF-Multicast

Recursively extend (a la fractal…) for flooding

Broadcast version…
Potpourri: Single-Route Problems…

Shortest path (GPSR…)

TBF

TBF+

a) Scenario illustrating three source-to-sink data-streams

b) In-network energy distribution - Region A

c) Energy distribution in the vicinity of the sink node - Region B
Multi-Path Routing

- Disjoint Paths
- Breaded Paths

Goals:
I: Ensure robustness (i.e., the network is not quite reliable)
   ⇒ Ship a packet along > 1 route

II: Prolong the lifetime (careful about the definition...!!!) by alternating the routes used by consecutive packet (possibly, in batches...)

Potpourri...
Beziers Curves

Sample: Cubic Bezier Curve
Beziers Curves **ARE** Rational Polynomials!

**Properties**

- End-point interpolation
- Convex hull
- Pseudo-local control
- **Affine invariance**

\[ C(u) = \sum_{i=0}^{i=n} B_{i,n}(u) \cdot P_i, \quad u \in [0,1] \]

✓ \( P_i \) are called **control points** of the generalized Bezier curve
Routing

Wake/Sleep Periods

- Remotely Activated Switch (RAS)
- Controlled by the MAC layer
- Usage of WAKE packets

Data-packet
WAKE packet
Field-Based Routing

How does one define “trajectory” in field-based settings?

- Or, for that matter, a collection of trajectories
  - Without too much overhead on their “construction”…

In this work:

- Electrostatic Field

\[
E(r) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \text{sgn}(i) \frac{|q_i|}{|r_i - r|^3} (r_i - r)
\]
Electrostatic Field-based Routing (EFR)

- Multipath for a (source, sink) pair:
  - Determine the “charges”
    - E.g., application-based priority
  - Specify the geo-locations
  - Let each node calculate the field-value (vector)
    - Use this to determine the next-hop…
    - Node which is nearest to the field-line (and towards the sink)
The distribution of nodes is hardly-ever:

- Uniform and dense-enough
- Uniform-enough along field line(s)

Two (and more) paths have merged into one

Completely oblivious to a possible “recovery”
Persistent EFR (Load-Aware)

- Add a little “memory” to the packets…
  - Determine the location of the 1-hop neighbors
  - Calculate the (unique angle of the tangent to the) field-curve passing through their location
  - Transmit that along with the rest of the packet…
Persistent EFR (Load-Aware)

- Enforce the “honor-the-ancestry”
  - Closest to the original curve (and towards the sink).

Next hop for the “red” packets

Next hop for the “green” packets

Much better “spread-out”
Persistent EFR (Load-Aware)

Additional concern:
- Boundary-effects (too many mergers)

Apply "method of images"
⇒ Add "virtual charges"

A couple of extra virtual-charge by the corners of the field...
Persistent EFR (Load-Aware)

- When **multiple sources** are supposed to sample and transmit data to a given sink:
  - Nodes “on the boundary” will have to decide on behalf of which source they transmit…
  - Need to carefully select the value of the charges of each source
Other “Esoteric” Routings (Curveball)

Fig. 3. The reversed stereographic projections of a grid network (9 × 9 grid in a 20 × 20 square area) to the sphere with various radii (2, 5 and 10).
Curveball Routing (Stereographic)

Bounding Property…

Fig. 4. The shortest distance between two points $m'$ and $n'$ on the sphere is the shorter segment of the greatest circle between $m'$ and $n'$.

Circular Sailing

Fig. 7. The Euclidean path length of proposed CSR protocol is bounded by the Euclidean path length of shortest path routing.
Background and Motivation (multipaths)

- **Multipath routing**
  - Uses *simultaneously* (but) distinct routes to transmit the *same* information
  - Robustness/Reliability

- **Alternating path routing**
  - Uses a *sequence* of distinct routes to transmit *new* (*well, “subsequent”) information
  - Load-balancing

- **Alternating Multi-paths**
  - Combines the strategies of the first two.
  - Robustness + Load-balancing
Background and Motivation (tributaries and deltas)

Much like in nature…

Original work (SIGMOD’05):
- In-network aggregate processing
- Reliability
  - When too many packets drop, “convert” a part of the tributary into a delta (and vice-versa)
- Demonstrated correctness/viability
Goal:
- Overall lifetime extension of the network

Trade-off:
- Latency vs. load balancing

This work:
- Explore the possibility and impact of combining *multiple trees* and *multiple-multipaths*
  for routing when processing a query with respect to a given region of interest
Basic Settings (Query Region and Answer Transmission)

- **One aggregation tree + multiple (k-short based) paths**
  - a.) Shortest multipath
  - 1st variant

- **An alternate tree + multiple (k-short based) paths**
  - b.) Alternative multipath + tree

- **Disjoint trees + sets of multiple (k-short based) paths**
  - c.) Multiple trees and multipaths
  - 2nd variant
Alternation of Tributaries and Deltas

Initialization steps:

I. Query specification:
   - the region of interest \( Q_R \),
   - the closest point to the sink \( B_c(b_{cx}, b_{cy}) \) on (the boundary of) \( Q_R \) for initial shortest path establishment
   - Additional tolerable delay bounds

II. Query propagation to \( B_c \)'s nearest node (\( N_{Bc} \))

III. \( N_{Bc} \) triggers a tree construction mechanism, constrained by \( Q_R \), rooted at \( N_{Bc} \)

Ex: boundaries for the roots of alternating trees
Alternation of Tributaries and Deltas (“construction”)

“Level$_i$ overlap” (parameter)

Given $Tr1(root1)$ and $Tr2(root2)$, where $root1\neq root2$, their $level$_$i$ overlap is the set of nodes that are at $level$_$i$ in both $Tr1$ and $Tr2$

Intuitively, it provides a measure for “spreading” between adjacent alternative roots for the purpose of balancing the load near the roots

Selection of alternating trees/roots:

I. Determine the boundaries of the possible locations of the roots for the alternating trees

II. Within these boundaries, find a set of nodes that do not violate, pairwise, the $level$_$i$ overlap
Concurrent Transmission with Disjoint Trees

Basic steps:

I. Partition $Q_R$ in two sub-regions with **balanced node count** (Ham-Sandwich cut):
   
   a. Color one region in “red”, the other in “blue”. “Red” region represent the admissible space of root nodes, with respect to the acceptable latency-parameter
   
   b. Bisect red/blue areas in half using a separator line ($O(n)$ for convex polygons)
Practical Considerations

- Avoid (or, minimize) *sharing of nodes* by both Tributaries and Deltas (load balancing)
- **Frequency** of alternating of trees/path should be carefully chosen
- The *sequence of alternating* among the trees/paths becomes important in high-sampling rate queries (queuing effect among adjacent routes can yield prolonged contention along routes)
Experimental Evaluation

Lifetime: *single vs. alternating* (k-short based) multipaths

Overall lifetime
Experimental Evaluation

Minimum residual energy depletion rate over the entire network (coincides, not surprisingly with the root nodes’ energy levels)