Geographic Routing

GPSR & GEAR

Sergei Vassilvitskii
Papers

- GPSR: Greedy Perimeter Stateless Routing for Wireless Networks
  - Brad Karp, H.T. Kung

- Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks
  - Yan Yu, Ramesh Govindan, Deborah Estrin
GPSR: Motivation

- Ad-hoc routing algorithms (DSR, AODV)
  - Suffer from out of date state (DV)
  - Torrents of link status messages (LS)

- Can use geographic information for routing
  - Assume every node knows position (x,y)
  - Keep a lot less state in the network
  - Require fewer update messages
GPSR: Algorithm

- Whenever possible, use Greedy forwarding.
  - Look at all neighbors, select closest to destination
- When all neighbors are further away
  - Use the right hand rule and walk around the obstacle
  - Resume greedy forwarding as soon as possible to deliver message to destination
- Note: no attempt is made to conserve energy
How to route around an obstacle?
- Locally discover a connected planar graph as subset of network graph.

- Edge (u,v) can be safely removed.
GPSR: Details (2)

- When stuck, use right hand rule to go around obstacles.

- Always go along the face that is crossed by xD
GPSR: Results

- GPSR delivered > 98% of packets successfully, similar to DSR
- Protocol overhead was 50% less: 15,000 vs. 30,000 packets
- GPSR used shorter paths.
- Heuristic results, no theoretical guarantees.
Papers

- GPSR: Greedy Perimeter Stateless Routing for Wireless Networks
  - Brad Karp, H.T. Kung

- Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks
  - Yan Yu, Ramesh Govindan, Deborah Estrin
GEAR: Motivation

- Automatic Load Balancing in Routing

Original Network
GEAR: Motivation (2)

- Automatic Load Balancing in Routing

Not Balanced Network
GEAR: Motivation (3)

- Automatic Load Balancing in Routing

Load Balanced Network
GEAR: Algorithm (1)

- **Assumptions**
  - Nodes know location, energy level of themselves and neighbors
  - Bidirectional links
  - Routing to a region R

- **Trick**
  - Maintain and update, a cost $h(N_i, R)$ for each neighbor & destination region.
GEAR: Algorithm (2)

- **Routing:**
  - Look at all neighbors closer to the destination, pick one with lowest learned cost.
  - If all neighbors are further away, pick the neighbor with the lowest learned cost, update the cost.

- **Initial Cost**
  - \( H(N, R) = \alpha \text{dist}(N, R) + (1-\alpha) \text{Energy}(N) \)

- **Cost Update Rule**
  - \( H(N, R) = h(N_{\text{min}}, R) + C(N, N_{\text{min}}) \)
GEAR: Example

- \( H(B,T) = 2, \ H(C,T) = \sqrt{5} \)

- Update \( H(B,T) = 2 + C(C,B) \).
- Next Round: \( S \rightarrow B \)
GEAR: Analysis

- ✓ Delivery Guarantee
- ✓ Load Balancing
- ✓ Routing around Holes

But
- Can get stuck in local minima while routing
- Basically implements Q-Learning (AI)
- $\Omega (n^2)$ Convergence time in worst case.
GEAR: Bad Case
GEAR: Results

- 50 to 100% increase over GPSR in number of packets transmitted before network partition.
- 50% increase over GPSR in number connected pairs after network partition.
- 25 to 45% increase in the average path length taken by a packet.
Conclusion

- Both rely on geographic information to speed up routing.
- Use different techniques to route around holes.
- GEAR tries to automatically load balance the network traffic when routing around holes.

- Any questions?