Rumor Routing Algorithm

Aleksi Ahtiainen
Aleksi.Ahtiainen@hut.fi

T-79.194 Seminar on Theoretical Computer Science
Feb 9 2005
Contents

• Introduction
• The Algorithm
• Research Results
• Future Work
• Criticism
• Conclusions
Introduction

• Rumor Routing Algorithm is described in paper:
  
Routing in Wireless Sensor Networks (WSNs)

- How to reach event nodes from the query node?
- Route consists of short hops
- Event is a localized phenomenon detected by some node(s)
- Query can be:
  1. A request for information
  2. Orders to collect more data
  3. Some unlocalized order, e.g. “Find a node with a camera and enough power to use it, and order it to take a photograph”
Challenges of WSN Routing

• Energy is in short supply
  - Use only short-distance message transmission
  - Minimize number of transmissions

• Wireless ad-hoc network with possibly failing nodes

• Often no common coordinate system available for the nodes
Traditional Routing [1/2]

- Event flooding:
  - When node detectd an event, it broadcasts information about it in its surroundings and other nodes repeat this
  - The nodes store the information, where they received the event from for later querying and/or the event is noticed by some monitoring query node
  - Transmission energy comparable to Event count * Node count

- Query flooding:
  - Query node broadcasts the query through the whole network
  - Transmission energy relative to Query count * Node count
Traditional Routing [2/2]

- Problems with flooding:
  - High energy consumption due to unnecessary transmissions
  - Message loss due to collisions caused by many simultaneous transmissions
- For example probabilistic broadcast has been suggested
- If geographical information is available, greedy shortest path algorithms can be used
Solution: Rumor Routing

- Main idea:
  - Agent messages precreate paths leading to event nodes as the events happen.
  - Later queries are sent on random walk until they find one of the paths, and then route along the path to event nodes.
When to Use Rumor Routing?

- Number of queries per event is high enough. If not, better to flood queries.
- Number of queries per event is low enough. If not, better to flood events.
- In best case: ~5..~36 queries per event
  - Small amount of data flowing back from event to query node. Otherwise cases better to find the shortest route by query flooding.
  - No coordinate system available. Otherwise greedy shortest path algorithms are better.
  - Each node has distinct identification number and knowledge of neighboring nodes
  - Nodes have similar transmission functionality (no hierarchy)
Algorithm - Basics

• Each node has
  - A neighbor list (generated when the network is initiated)
  - An event table with forwarding information to events it knows of
    • Possibly timestamped for expiration
Agents [1/4]

• When a node detects an event it:
  => stores a path of distance zero to the event in the node
  => creates an agent probabilistically:
    - reason for using probability: usually many nodes notice the same event

• Agent travels for some maximum amount of hops

• Agent contains an event table and combines it with event tables in visited nodes
Agents [2/4]

• Agents aggregate paths
Agents [3/4]

- Agents optimize longer paths
Agents [4/4]

- Agents use a straightening algorithm:
  - Record recently seen nodes and avoid travelling to them if possible
- Neighboring nodes often overhear messages not sent directly to them and can use the information to optimize paths
  - So in fact the paths created by agents are thick trails
Routing Queries

- Query also has some maximum number of hops
- First random walk, then along the path
- If destination was not reached, the query node can either retransmit or flood
- Straightening algorithm used in the random walk
- Possible loops in agent paths can be avoided:
  - Use random ids for queries,
  - store recently seen query ids in nodes and
  - when nodes receive a query on the list, they send it in random direction instead of along the path
Research Setting

- Paper describes simulation results
- 200x200 m$^2$ 2-dimensional area with node communication radius of 5 meters
- 3000-5000 randomly scattered nodes
- All events also at 5m-radius circles
- Precreated event distribution (10-100 events) and agent paths
- After that 1000 queries to random events from random query nodes
- Queries flooded after first failure
- Different agent and query hop counts tested
Research Results [1/2]

- With minimal setup costs (small agent hop count and less than 25 agents) only 60% of queries successfully delivered. Even query flooding would have been better.

- With high setup costs (over 400 agents) algorithm had setup costs higher than event flooding, but the query routing success was 99.9%

- Best settings: Small number of agents (31 for 10 events) and high agent maximum hop count (1000), 98.1% of queries were delivered with average energy of 1/40th of query flood. Setup cost was was then equal to about 8 query floods.
  - Rumor routing better than flooding when queries per event between 5 and 36
Research Results [2/2]

- Algorithm had stable results over several test runs
- But the guaranteed query delivery rate depended heavily on the random distribution of events and queries, i.e. it is difficult to guarantee some energy use for real-life situations
- Fault-tolerant up to 20% node failures, above this strong performance loss
Future Work [1/2]

- **Network dynamics and asynchronous events**
  - In reality events occur in time and algorithm is likely to favor older events

- **Collisions**
  - Rumor routing is likely to suffer less from collisions than flooding algorithms

- **Non-localized events.**
  - How are queries like “find a node with a camera and enough power” handled

- **Non-random query pattern**
  - Often queries are generated by base-stations or in some networks by nodes close to the actual events
Future Work [2/2]

- Non-random next hop selection in the algorithm
  - If some localization information is available, agents could leave behind information on already visited regions and other agents could later try to cover these

- Use of constrained flooding
  - Instead of random walk, queries could first be flooded at a short distance. Problem is then, how to decide which queries to forward

- Parameter setting exploration
  - Optimal parameters depend heavily on the event and query patterns, perhaps the algorithm could somehow configure itself on the fly
Criticism

- The authors do not describe any method (except brute force) for finding good parameter values
- Test settings and results are not described very thoroughly
Conclusion

- Rumor routing is a good and tunable algorithm for many situations, in which flooding would generate too much traffic and geographic information is not available.