Summary of Rumor Routing in Wireless Sensor Networks

Aleksi Ahtiainen
Helsinki University of Technology
Laboratory of Information Processing Science
P.O.Box 5400, FI-02015 TKK, Finland
Aleksi.Ahtiainen@tkk.fi

Abstract

Rumor routing is a wireless sensor network routing algorithm, which aims at lower energy consumption than algorithms that flood the whole network with query or event messages. The algorithm is tunable and its usefulness depends on how well the configuration parameters are set for the particular event and query distribution in the network. The algorithm also handles node failures and allows for tradeoffs between setup overhead and delivery reliability. This paper summarizes the algorithm as it is described in [1].

1 Introduction

Wireless sensor network is usually a network with thousands of randomly scattered simple sensor nodes with limited data communication capability. Energy is in limited supply and therefore routing queries to events in the network energy-efficiently has high priority. Since long-range transmissions consume a lot of energy, routing must be based on short hops between communicating nodes. The number of these short hops must also be minimized. Rumor routing algorithm described by David Braginsky and Deborah Estrin in their paper "Rumor Routing Algorithm For Sensor Networks" [1] is one solution to the problem.

Main idea of rumor routing is to create paths leading to each event when the event happens, and later to route queries along these paths. In order to join the path, the queries are first sent on a random walk in the network.

In the text events are assumed to be any localized phenomena detected by the network. Queries in turn can be requests for information or orders to collect more data, or even something unlocalized, e.g. "Find a node with a camera capability and enough power to use it".

Rumor routing is only applicable and beneficial in some situations. Other alternatives are usually better, when

- the amount of data flowing back from event node to query node is significant. In such cases it is better to flood the query messages through the network in order to find the shortest path between the query and event nodes.

- the amount of queries per event is high. In such cases it is usually better to flood messages from event nodes through the whole network.
- the nodes have established a common coordinate system. Then greedy shortest path algorithms are usually better.
- nodes don't have distinct identification numbers or knowledge of their neighboring nodes' identifications. Then flooding needs to be used.
- nodes have a hierarchy of different transmission abilities.

Rumor routing's beneficial range between two thresholds of number of queries per event is demonstrated in figure 1.

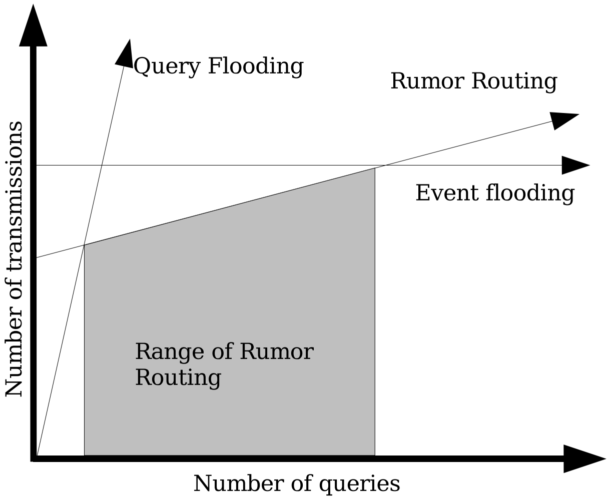


Figure 1. Rumor Routing Range [1]

2 The algorithm

2.1 Basics

The basic idea of rumor routing is to use agents to create paths leading to each event when the event happens. The agents are actually long-lived messages traversing in the network. Later queries can be routed along these agent-generated paths. In order to join the path, the queries are first sent on a random walk in the network. See figure 2.

Each node in the network maintains a list of its neighbors and an event table with forwarding information to all the events it knows of. When the network is initiated, the neighbor lists are generated by broadcasting each node's id and listening to the broadcasts. If the events are only needed for a certain amount of time or the size of the event table is limited, expiration timestamps can be added to the event table entries.

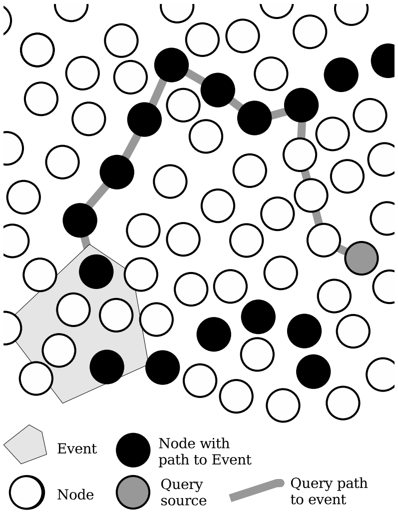


Figure 2. Query's path [1]

2.2 Path-creating Agents

The paths are stored as states in single nodes and created by travelling agents. The agents are created in event nodes by adding a route of length 0 to the event and probabilistically generating an agent. The probability is used because usually many nodes notice the same event and too many paths to the same event generates too much overhead.

The agent travels in the network for some maximum number of hops. On its way it combines its own event table with event tables of visited nodes. Whenever an agent crosses a path leading to another event, it starts to create an aggregate path to both (or multiple) events. See figure 3. Also when the agent finds a node with longer path than its own to the same event, it updates the routing table with the shorter path. See figure 4.

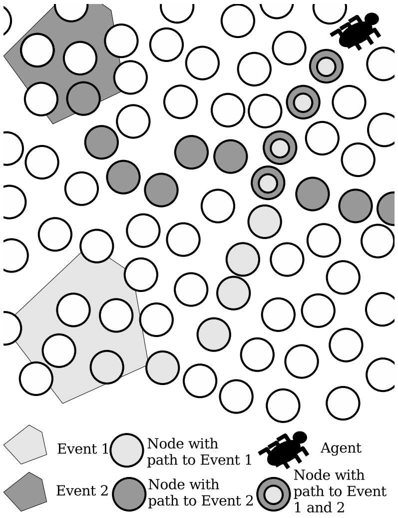
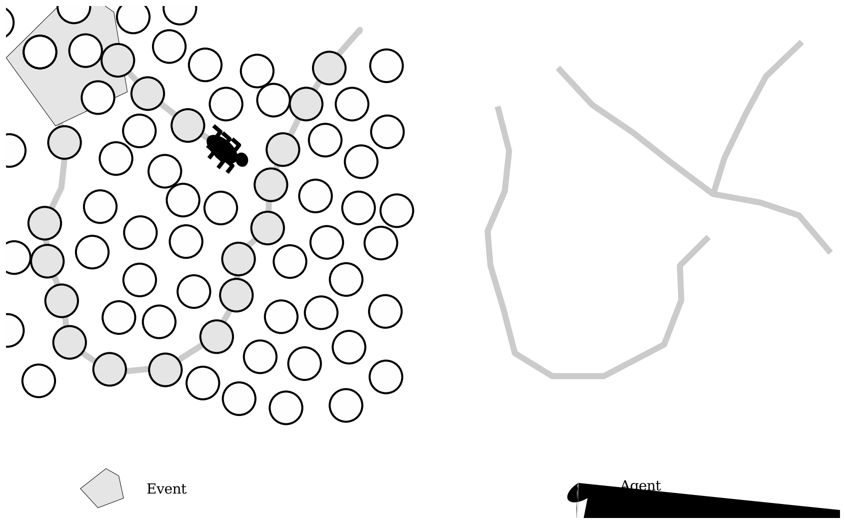


Figure 3. Aggregating two event paths [1]


Event
Agent

through the network, which guarantees delivery. Then the energy needed for routing q queries is:

$$E_t(q) = E_{setup} + q * (E_{path} + N * \frac{Q_{total} - Q_f}{Q_{total}})$$

, where $E_t(q)$ is the amount of energy needed for routing q queries, E_{setup} the amount of energy needed for setting up event paths by agents, E_{path} the average energy needed along an agent-generated route, N number of nodes, Q_{total} the total number of queries in the test run and Q_f the number of queries that needed flooding.

In comparison, with query flooding:

$$E_t(q) = q * N$$

and event flooding:

$$E_t(q) = E * N$$

The simulations were run with N = $\{$3000,4000,5000$\}$ nodes scattered randomly on a two-dimensional field of $200 * 200m^2$. Each node had transmission capabilities in a 5m-radius circle. The simulation pregenerated E = $\{$10,50,100$\}$ events of 5m-radius circle and after that Q = 1000 queries originating in random nodes. For every combination the following parameters were tested:

- actual number of agents created (recorded, actually set up by setting probability)
- maximum hop count of agents (100, 500, 1000 hops)
- maximum hop count of queries (1000, 2000 hops)

With minimal setup costs (agent maximum hop count only 100 with small number of about 25 agents), only 60% of the queries were delivered. A high number of agents (around 400) had too high setup cost (above event flooding), but then the query routing success was 99.9%. For a wide variety of settings between these rumor routing was better than event flooding.

Best results were found with small number of agents (31), and high agent maximum hop count (1000). Then 98.1% of queries were delivered with an average cost of 92 cumulative hops per query (only 1/40th of query flood). Setup cost was equal to about 8 query floods. With queries per event running from 5 to 36, rumor routing performed better than query or event flooding.

The algorithm performed better than event flooding up to a certain event cost threshold with most of parameter values. With higher number of events (E) and nodes (N), the threshold increased, due to the fact that the cost of event flooding is $E_t(q) = E * N$.

The algorithm was found to be stable at least in this configuration by counting averages and standard deviations over several runs.

The guaranteed delivery rate depended heavily on the event/node/query distribution and also had a high standard deviation. In 100% of cases guaranteed delivery rate was 62%, and in 50% of cases 90% delivery rate was reached. The mean was 85% with high standard deviation of 8.8%. In practice this means, that when networks are deployed, it is difficult to guarantee some query delivery rate due to the randomness of event/node/query distribution.

The algorithm was quite fault tolerant up to 20% node failure. Above this performance degrades severely. Delivery rate was strongly correlated to the number of failed nodes, with correleation coefficient of 0.91. With 5% node failure 90% of the queries were still delivered successfully.

4 Future Work

The authors identify some issues, that would need further development and testing:

- Network dynamics and asynchronous events. In reality events occur in time, in current simulation all events were pre-generated. The algorithm is likely to favor older events, because more paths are created for them.
- Collisions. Rumor routing is likely to suffer less from communication collisions than flooding-based algorithms.
- Non-localized events. Like the example of routing queries to nodes, that have a camera or some other capability and enough power to use it.
- Non-random query pattern. In practise queries are often generated by single base station nodes, instead of random nodes like in the used simulations. Also in some cases the query nodes are likely to be close to the queried event nodes.
- Non-random next hop selection in the algorithm. If localization information is available, agents could try to leave information about already explored regions in the network. Then other agents could try to reach non-explored regions instead of travelling randomly.
- Use of constrained flooding in the algorithm. Instead of routed randomly, the queries could be flooded for a short distance. The problem then is the selection of which queries to forward, since there are many simulataneus discoveries of agent-generated paths.
- Parameter setting exploration. The optimal parameters depend heavily on the event and query patterns. There might be a way to use the network itself to find best parameter values and set them on the fly.

5 Conclusion

Rumor routing is a tunable and more energy-efficient algorithm than flooding-based ones in many situations, especially when geographic information is not available. It also handles node failures quite well. Rumor routing is a good choice also when events are not geographically locatable, like large concentrations of some chemical or looking for some acoustic pattern in a big network.

References

[1] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM Press, 2002.