Rumor Routing Algorithm for Sensor Networks

David Braginsky, Deborah Estrin

EE228A presentation
Design Considerations and Requirements

- Each node has limited computational power
- Each node has limited communication capabilities

Network:
- Self-configuring, scalable, redundant
- Robust to shifting topologies
- Reasonable failure rates
Describe a method of routing queries to nodes that have observed a particular event, which allows to retrieve the data keyed on the event and not the underlying network addressing scheme or geography.
Definitions

• **Event**
  – abstraction identifying anything from sensor readings to processing results
  – assumed to be localized occurring in a fixed region of space

• **Query**
  – request for information
  – order to collect specific data
Query is originated from the query source and searches for path to the event. As soon as it finds the node with event path info it is routed directly to the event.
When agent creating the path to the green event comes across a path to the red event it continues propagating the aggregate path to both
Agents also optimize the path to the event
Rumor Routing Algorithm

- Network consists of densely distributed wireless sensor nodes with relatively short symmetric radio range
- Nodes record events and route queries
- Each node maintains a list of neighbors, list of events and forwarding information to all the events it knows
- Neighbors list is created and maintained actively by broadcasting a request or passively by listening to other node broadcasts
Agents

- Carry the list of events
- Inform nodes of the known events and routes to them
- Perform sync of routing tables
- Maintain the list of recently seen nodes
- Pick the next hop not from that list
- Created upon event
- Die when TTL expires
Queries

- Can be generated any time by any node
- Have reasonable TTL
- Perform random walk until find path to the event
- Maintain the list of recently seen nodes to avoid loops
- If still undelivered, then they can be retransmitted or flooded
Flooding Mechanisms

• Query Flooding
  – Independent of the number of events
  – Useful if the number of events is very high compared to the number of queries

• Event Flooding
  – Independent of the number of queries
  – Useful if the number of queries is very high compared to the number of events
When to use Rumor Routing

- Number of Queries
- Number of Transmissions

- Query Flooding
- Event Flooding

Rumor Routing - one possibility
Simulation

http://lecs.cs.ucla.edu/~daveey/art/code.html

- Area: 200x200 sq. meters
- Number of nodes: $N = \{3000, 4000, 5000\}$
- Neighbors: nodes within 5m from each other
- Number of queries: $Q$
- Energy for query flooding: $N \times Q$
- Number of events: $E = \{10, 50, 100\}$
- Energy for event flooding: $N \times E$
- Number of delivered queries: $Q_d$
- Average energy per query: $Eq + N \times (Q - Q_d)/Q$, where $Eq$ – average energy spent on routing a query
- Energy for rumor routing: $Es + Q \times (Eq + N \times (Q - Q_d)/Q)$, where $Es$ – energy required to establish routes
Tunable Parameters

- Number of Agents: $A$
- Agent TTL: $La$
- Query TTL: $Lq$
Simulation Results

100 Events, 4000 Nodes

- Query Flooding: \(A=589, La=1000, Lq=2000\)
- Event Flooding: \(A=31, La=1000, Lq=2000\)

Number of Transmissions (thousands)

Number of Queries
Simulation Results

10 Events, 4000 Nodes

Number of Transmissions (thousands) vs. Number of Queries

Query Flooding

Event Flooding

A=28, La=500, Lq=1000

A=52, La=100, Lq=2000
TODO

- Network Dynamics: events do not occur simultaneously
- Consideration of Collisions
- Non-localized Events
- Query Pattern and Next Hop Selection
- Parameter Settings
Conclusion

Algorithm may be practical under certain conditions, but when and how – TBD. In other words, there is still a lot of work ahead.