A COMPARATIVE STUDY OF ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

Adam Chalak, Vijay Sivaraman
Telecommunications Engineering
University of New South Wales
Sydney, Australia
email: {z3020006@student.unsw.edu.au,vijay@unsw.edu.au}

Nevin Aydin
Department of Computer Engineering
Istanbul University
Istanbul, Turkey
email: nevin.aydin@gmail.com

Damla Turgut
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, USA
email: turgut@cs.ucf.edu

Abstract–**Due to the limited processing power, and finite power available to each sensor node, regular ad hoc routing techniques cannot be directly applied to sensor networks domain. Thus, energy-efficient routing algorithms suitable to the inherent characteristics of these types of networks are needed. Routing algorithms must also be robust to failures, and provide low latency. This paper makes a performance comparison of three sensor network routing protocols, namely, Rumor routing, Stream Enable Routing (SER) and SPIN. The results show that SPIN is the most suitable for small size networks while SER serves the large scale networks the best. Rumor is considered an alternative protocol with high delivery rate and scales from small to medium size networks.**

Index Terms–**sensor routing, performance comparison**

1 Introduction

A wireless sensor network consists of light-weight, low-power, small size of sensor nodes. The areas of applications of sensor networks vary from military, civil, healthcare, environmental to commercial. Example applications include forest fire detection, inventory control, energy management, surveillance and reconnaissance, and so on. Due to the low-cost of these nodes, the deployment can be in order of magnitude of thousands to million nodes. The nodes can be deployed either in random fashion or a pre-engineered way. The sensor nodes perform desired measurements, process the measured data and transmit it to a base station, commonly referred to as the sink node, over a wireless channel. The base station collects data from all the nodes, and analyzes this data to draw conclusions about the activity in the area of interest. Sinks also can act as gateways to other networks, a powerful data processor or access points for human interface. They are often used to disseminate control information or to extract data from the network.

Nodes in sensor networks have restricted storage, computational and energy resources; these restrictions place a limit on the types of deployable routing mechanisms. Additionally, ad hoc routing protocols, for conventional wireless networks support IP style addressing of sources and destinations. They also use intermediate nodes to support end-to-end communication between arbitrary nodes in the network. It is possible for any-to-any communication to be relevant in a sensor network; however this approach may be unsuitable as it could generate unwanted traffic in the network, thus, results the extra usage of already limited node resources. Many-to-one communication paradigm is widely used in regards to sensor networks since sensor nodes send their data to a common sink node for processing. This many-to-one paradigm also results in non-uniform energy drainage in the network.

There are already many existing routing protocols in wireless sensor networks. These protocols can be grouped into two main categories, i) single path algorithms [5] and location aware routing algorithms [7]. Single path algorithms require the sink node to flood the network periodically to discover new routes to redirect traffic around the failed nodes. At first, this approach may sound unsuitable for sensor networks since flooding the network requires high energy consumption, resulting the shortened network lifetime. Location aware algorithms on the other hand require each node to know its geographical location with the help of GPS for instance. Earlier, GPS was not considered usable in all types of networks since it does not work indoors or under dense foliage, but recent discoveries suggest this may have been overcome [9].

In this paper, the performance of Rumor [2], SER [1], and SPIN [4] have been compared. These algorithms are simulated for various settings, with a view of identifying the most suitable protocol for different applications. The rest of the paper is organized as follows. Brief summaries of these protocols are given in the next section. Section 3 gives detailed performance study while the paper is concluded in Section 4.

2 Background

In this section, the three algorithms, namely, Rumor, SER, and SPIN are discussed.

2.1 Rumor Routing

Rumor routing [2] allows the routing of queries to nodes that have observed an event of interest. As a result, retrieval of data is based on events and not on an addressing scheme. An event is an activity related to the phenomena being sensed (e.g. increased movement in an area being monitored). In this paper, events are assumed to be localized phenomena which occur in fixed regions of space. A query is issued by the sink node for one of two reasons, as an order to collect more data, or as a request for information. Once a query arrives at its destination, data is issued to the originator of the query. Depending on the amount of data (whether it is more or less) being issued to the originator of the query, shorter paths from the source to the sink are discovered.

Various methods have been proposed to find shortest paths, including, flooding the query through the network. Directed diffusion [3] is such an example; however directed diffusion resorts to flooding the query throughout the network in order to find the best path, while Rumor routing can find the best path using other methods, and only resorts to flooding as a last choice.

If flooding was to happen on a regular basis, network resources would be consumed quickly, thus Rumor routing was created to be an alternative to flooding queries and events. When a query is generated, it is sent randomly through the network until it finds the event path instead of flooding it. When the query finds the event path, it is routed directly to the event. Only if the path cannot be found, it is flooded as a last resort. Rumor routing can achieve a high delivery rate as will be shown in the performance study.

Rumor routing uses agents, which have a limited life determined by a TTL field; these agents create paths in the direction of any events they may come across. If an agent crosses a path to an event that it has not yet come across in the network, it creates a path that leads to both events.

2.2 Stream Enabled Routing

Stream Enabled Routing (SER) [1] allows the source nodes to choose routes based on instructions given to it by the sink node. An important feature of SER is that it takes into account the available energy of the sensor nodes. Also, SER allows the sink to give new instruction to the sources without setting up another path, as a result conserving valuable network energy resources.

SER requires sink nodes to specify the sensor nodes that perform the tasks in their instructions. If the nodes do not have a GPS, a location aware protocol, such as [6] can be used to approximate their locations. One of the advantages is that it can be integrated with the application layer easily since it is based on instructions and tasks. An instruction is defined as an identifier value. This conserves memory because only the identifier is sent rather than the whole attribute list. There are four types of messages that are sent through the network, information message (I-message), scout message (S-message), neighbor-neighbor message (N-message), and update message (U-message). The S-message is broadcast for the sources to select routes between themselves and sinks based on the quality of service requirements of the instructions. SER also takes into account the memory limitations of nodes, energy of nodes, and the QoS of the instruction. After the routes are established, the sink node can give new instructions to the sources without setting up another route.

2.3 Sensor Protocols for Information via Negotiation

Sensor Protocols for Information via Negotiation (SPIN) [4] are a family of protocols used to efficiently distribute information in a wireless sensor network. Conventional data dissemination approaches such as flooding and gossiping waste valuable communication and energy resources by sending redundant information throughout the network. In addition, these protocols are not resource-aware or resource-adaptive. SPIN solves these shortcomings of conventional approaches using data negotiation and resource-adaptive algorithms. Nodes running SPIN assign a high-level name to their data, called *meta-data*, and perform meta-data negotiations before any data is transmitted. This assures that there is no redundant data sent throughout the network. In addition, SPIN has access to the current energy level of the node and adapts the protocol it is running based on the remaining energy. Simulation results show that SPIN is more energy-efficient than flooding while distributing data at the same rate or faster [9], however as we will show in section III, Rumor routing still outperforms SPIN. The SPIN family of protocols uses three messages for communication.

- **ADV:** When a SPIN node has some new data, it sends an ADV message to its neighbors containing meta-data (data descriptor)
- **REQ**: When a SPIN node wishes to receive the data, it sends an REQ message
- **DATA**: These are actual data messages with a meta-data header.

The SPIN family of protocols is made up of four protocols, SPIN-PP (a three–stage handshake protocol for point-to-point media), SPIN-EC (SPIN-PP with low-energy threshold), SPIN-BC (a three–stage handshake protocol for broadcast media), and SPIN-RL (SPIN-BC for lossy networks).

3 Performance Study

In this section, we present details of the performance study, including the simulation parameters used in simulation environment, and the results.

3.1 Simulation environment and parameters

LecsSim [8] simulator is used to conduct the performance assessments. LecsSim is designed to facilitate the testing of various distributed algorithms. It allows the user to create nodes, whose behavior is defined in a C++ class, arranged in a 2D topology. The nodes communicate by passing events to each other. Certain events are constrained by a propagation model, which can also be defined in a class.

The simulation parameters used are: (i)*agents per event*—the amount of agents generated per event. An agent's basic purpose is to travel around the network, constantly updating nodes' routing tables with the shortest route available to a destination; (ii) *agent TTL*—agents have a TTL field, that limits the lifetime of the agent in the network, hence preventing indefinite looping of agents; (iii) *query cycle*—nodes generate queries which target events; these queries circulate in the network. When a node in the network receives a query, it checks to see if it has a route towards the target event, which is specified in the query. If there is a route, it forwards the query along the path. Otherwise, it sends the query to a random neighbor. Every time a node forwards the query, the query's TTL field is reduced, such that the query will be dropped when this value reaches zero.

We performed Rumor routing simulations in LecsSim on networks of size of $100, 200, 400, 600, 1000$ nodes respectively. These nodes were scattered randomly on a $200m \times 150m$ 2D field. The placement of the nodes in the area was random rather than pre-determined locations. A more realistic propagation model was used, where each node, depending on its power levels, could send packets to any node within $2m$ to $5m$ from itself; $2m$ when low on power and $5m$ when high on power. This propagation model is realistic in the sense that there is no assumption that a sensor node can transmit data at a constant range regardless of power levels. A fixed event map was then generated, randomly placing $5, 50, 100, 200, 500$ events across the 2D field. A query pattern of 100 queries was then, generated from a random node to a random event.

As the nodes were initialized, they started to generate agents which setup paths, the query pattern was then run, and the number of successful routed queries was recorded. The performance data in this section is collected over 50 simulation runs.

3.2 Simulation Results

The network was flooded with queries to guarantee high delivery rate; however, additional $N \times (1000 - Q_f)$ sends were performed, where Q_f is the number of delivered

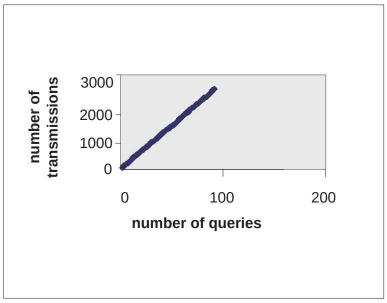

Figure 1. Simulation of 200 nodes with 5 agents per events. This resulted in a 97.9% delivery rate.

queries. The average energy used for each query (in a network of 1000 nodes) [2] was

$$\Big(E(q) + N \times (\frac{1000 - Q_f}{1000})\Big)$$

where $E(q)$ is the energy spent routing the queries.

The average energy per query and the setup energy can be used to find the total energy utilized by the network to route Q queries [2] as follows:

$$E = E(setup) + Q\Big(E(q) + N \times (\frac{1000 - Q_f}{1000})\Big)$$

E was set at $10, 50$ and 100 events. The Agent TTL and Query TTL remained constant. The agents per event were set for the values of $5, 10, 50$ and 100. Varying this value resulted in findings that the lower the number of agents per event, the less failures and dropped queries occurred. But this also meant that there were less queries sent to each node, showing that the fewer agents there are in the network, the less likely it is for other nodes to be aware of data being collected in the network. It is shown in Figures 1 and 2 that there were fewer failures and fewer queries in the network when the agent per event value was set at 5, as opposed to 100. The reason being that, more agents would mean processing more information for the nodes; hence draining their power. Similar outcomes resulted for other simulations as well.

Although with the agent per event value set at a higher value (100 or above), the data would return to the sink faster only with the expense of the network experiencing more congestion and failures. Intuitively, the increase in queries throughout the network also decreases the battery life of the nodes. It should be noted that since Rumor routing uses data dissemination to send data from sources to sink, the energy of the network is depleted faster than some other protocols.

Table 1 presents the parameters used in the simulations to determine delivery rates. The parameters given are found to obtain the optimum possible delivery rates for particular size networks. Although there is no set formula to determine the optimal values to use, Rumor routing has the ability to *tune* to a variety of different applications and network sizes.

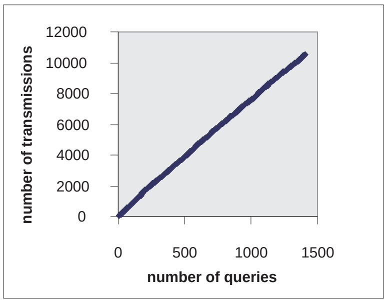

Figure 2. Simulation of 200 nodes with 100 agents per events. However, this resulted in only a 90% delivery rate.

Network size	Agents per event	Agent TTL	Query cycle	Delivery rate (%)
100	10	57	70	97.2%
200	18	30	24	97.9%
400	15	73	50	97.3%
600	28	78	80	97.2%
1000	31	1000	80	98.3%

Table 1. Simulation parameters values

It is important to compare the number of participating nodes in routing messages from source to sink for each of the routing protocols at hand. Since the lower amount of nodes participating in the routing would mean the lower the energy depletion of the network. From Figure 3, it is shown that SPIN has used 1000 nodes to send data from source to sink, while Rumor and SER used only 680 and 30 sensor nodes respectively [1]. We can also conclude that SPIN may not be suitable if the aim is to deploy the sensor network for long periods of time since the energy of the network would be depleted much faster. From these results, Rumor routing would work the best from small to medium scale networks.

Another important feature of any routing protocol is the time it requires to send a data from the source to the sink (see Figure 4). The shortest time was achieved with Rumor routing although jitter could not be measured directly for this protocol. The data has reached to sink on the average of 0.39 seconds in Rumor routing, while the SER protocol takes 0.73 seconds with about 0.02 seconds of jitter and SPIN takes 2.15 seconds for data to reach the sink [1]. The results produced by the Rumor routing may possibly vary if the jitter can be properly measured.

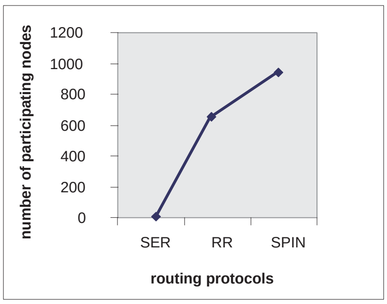

Figure 3. The average number of nodes participating in various routing protocols

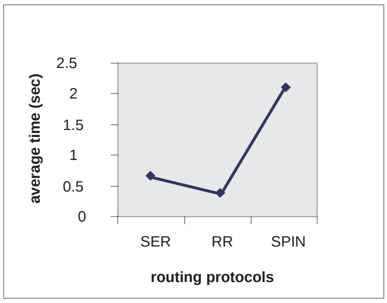

Figure 4. The average travel time of data from the source to the sink

4 Conclusions

We compared the performances of three routing protocols, namely, Rumor routing, SER, and SPIN. SER is a protocol particularly suited to large scale networks due to its excellent efficiency, latency and jitter properties. The fact that SER does not require nodes to have unique IDs further strengthens the argument of its suitability to large scale networks. SPIN was found to perform better in smaller size networks because of its efficiency and high latency properties. The use of SPIN in large scale networks could potentially exhaust system resources in a much faster pace. Rumor routing is considered an alternative protocol to the various flooding protocols presented. The results have shown that it is an efficient protocol with a high delivery rate. It was also concluded that Rumor routing may be most suitable for networks with small to medium in size.

References

[1] Akyildiz, I. F., and Su, W. "A Stream Enabled Routing (SER) Protocol for Sensor Networks", *Med-hoc-Net 2002*, September 2002.

[2] Braginsky, D. and Estrin, D. "Rumor Routing Algorithm for Sensor Networks", *WSNA 2002*.

[3] Estrin, D., Intanagonwiwat, C., and Govindan, R. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *MobiCOM 2000*.

[4] Heinzelman, W.R., Kulik, J., and Balakrishnan, H. "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Mobicom 1999*, pp. 174-185.

[5] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. "System architecture directions for networked sensors", *ACM ASPLOS IX*, 2000.

[6] Savvides, A., Han, C., and Srivastave, M. "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors", *MobiCOM 2001*, pp. 166-179.

[7] Yu, Y., Govindan, R., and Estrin, D. "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks", UCLA CSD *Technical Report UCLA/CSD-TR-01-0023*, May, 2001.

[8] "LecsSim-Wireless Network Simulator", http://sourceforge.net/projects/lecssim

[9] http://www.wherifywireless.com