Data dissemination and gathering in sensor networks

Dmitrij Lagutin, dlagutin@cc.hut.fi

1. Introduction

Sensor networks [1] contain a large number of sensor nodes that collect some kind of data, for example temperature or air pressure, from the environment. Even though both wireless ad hoc networks and sensors networks communicate wirelessly, they are quite different from each other. The number of nodes in sensor networks can be several orders of magnitude larger than in ad hoc networks. Sensor nodes are usually quite small, and they have very limited battery power. Their batteries are often not replaceable or rechargeable. Finally, sensor nodes may lack an unique identifier like an IP address that is always present in ad hoc networks.

These differences present challenges for data dissemination and gathering in sensor networks. Data dissemination and data gathering algorithms must take into account decentralized nature of sensor networks and limited battery power of sensor nodes.

2. Data dissemination

A data dissemination is a process by which data and queries for data are routed in the sensor network. In a scope of data dissemination, a *source* is the node that generates the data and an *event* is the information to be reported . A node that is interested in data is called *sink* and the *interest* is a descriptor for some event that node is interested in. Thus, after source receives an interest from the sink, the event is transferred from the source to the sink. As a result, data dissemination is a two-step process. First, the node that is interested in some events, like temperature or air humidity, broadcasts its interests to its neighbors periodically. Interests are then propagated through the whole sensor network. In the second step, nodes that have the requested data, send data back after receiving the request. Intermediate nodes in the sensor network also keep a cache of received interests and data.

There exist several different data dissemination methods. In this paper flooding, gossiping, SPIN [2], and cost-field approach [3] are covered in more detail.

2.1 Flooding

In flooding method each sensor node that receives a packet broadcasts it to its neighbors assuming that node itself is not the destination of the packet and the maximum hop count is not reached. This ensures that the data and queries for data are sent all over the network.

Flooding is a very simple method, but is has several disadvantages [2]. In flooding duplicate messages can be sent to the same node which is called implosion. This occurs when a node receives the same message from several neighbors. In addition, the same event may be sensed by several nodes, and thus when using flooding, neighbors will receive duplicate reports of the same event, this situation is called overlap. Finally, many redundant transmissions occur when using flooding and flooding does not take into account available energy at sensor nodes. This wastes a lot of network's resources and decreases the lifetime of the network significantly.

2.2 Gossiping

Gossiping method is based on flooding, but node that receives the packet forwards it only to a single randomly selected neighbor instead of sending it to all neighbors. The advantage of gossiping is that it avoids the problem of implosion and it does not waste as much network resources as flooding. The biggest disadvantage of gossiping is that since the neighbor is selected randomly, some nodes in the large network may not receive the message at all. Thus, gossiping is not a reliable method for data dissemination.

2.3 SPIN

Sensor Protocols for Information via Negotiation (SPIN) use negotiation and resource adaption to address the disadvantages of basic flooding. SPIN uses data-centric routing, nodes are advertising their data and they will send the data after receiving a reply from interested nodes.

SPIN uses three types of messages: ADV, REQ, and DATA. The sensor node that has collected some data sends an ADV message containing meta-data describing the actual data. If some of node's neighbors is interested in the data, the neighbor sends a REQ message back. After receiving the REQ message, the sensor node sends the actual DATA. The neighbor also sends ADV message forward to its neighbors, regardless wherever or not the neighbor is interested in data, thus data is disseminated through the network. Figure 2 below describes ADV-REQ-DATA exchange of SPIN.

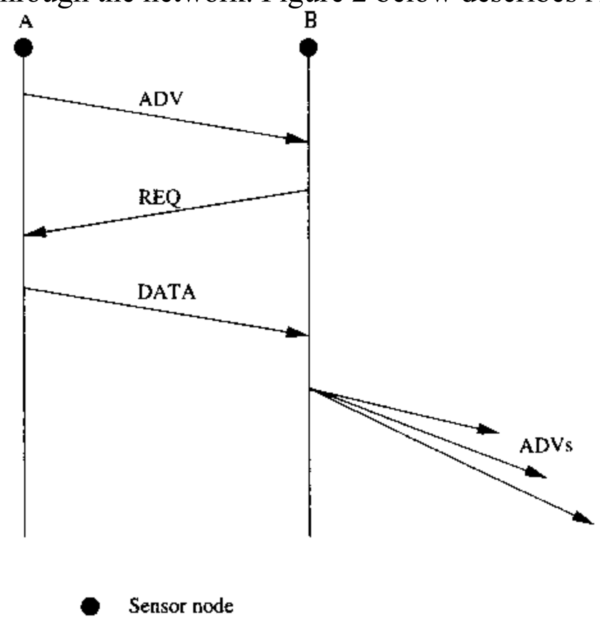

Figure 1: An example of SPIN (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

In the figure, node A advertises its data using an ADV message, its neighbor node B replies with a REQ message and thus node A sends actual data to the B. Node B also forwards ADV messages to its neighbors. Improved version of SPIN, SPIN-2 uses an energy or resource threshold to reduce participation of nodes. Thus, only those nodes that have sufficient amount of resources participate in ADV-REQ-DATA exchange.

SPIN is more efficient than flooding since the negotiation reduces the implosion and overlap. Resource adaptation in SPIN-2 prolongs the lifetime of the network: sensor nodes with low resources do not have to participate in ADV-REQ-DATA exchange and as a result they can collect data for a longer time.

2.4 Cost-field approach

The aim of the cost-field approach is to solve problem of setting paths to the sink. The cost-field approach is a two-phase process, first the cost field is set up in all sensor nodes, based on some metric like delay. In the second phase, data is disseminated using the costs. The cost at each node is the minimum cost from the node to the sink, which occurs on the optimal path. With the cost-field approach explicit path information does not need to be maintained.

In the first phase of cost-field approach, a cost field is set up starting from the sink node. The sink node broadcasts an ADV packet with cost set to 0. When node *N* receives an ADV packet from node *M*, it sets its own path cost to $min(L_N, L_M+C_{NM})$, where L_N represents the current total path cost from sink to node *N*, L_M is the cost from node *M* to sink, and C_{NM} is the cost from node *N* to *M*. When forwarding cost to other nodes in the network, cost-field approach uses back-off timers to avoid transmission of non-optimal costs. Otherwise cost-field approach would resemble flooding which is very ineffective. This means that node *N* will broadcast received ADV message forward only after time $\gamma * C_{NM}$ has passed, γ is the parameter of the algorithm. Figure 2 below shows an example of setting up the cost field and it also describes how back-off timers work.

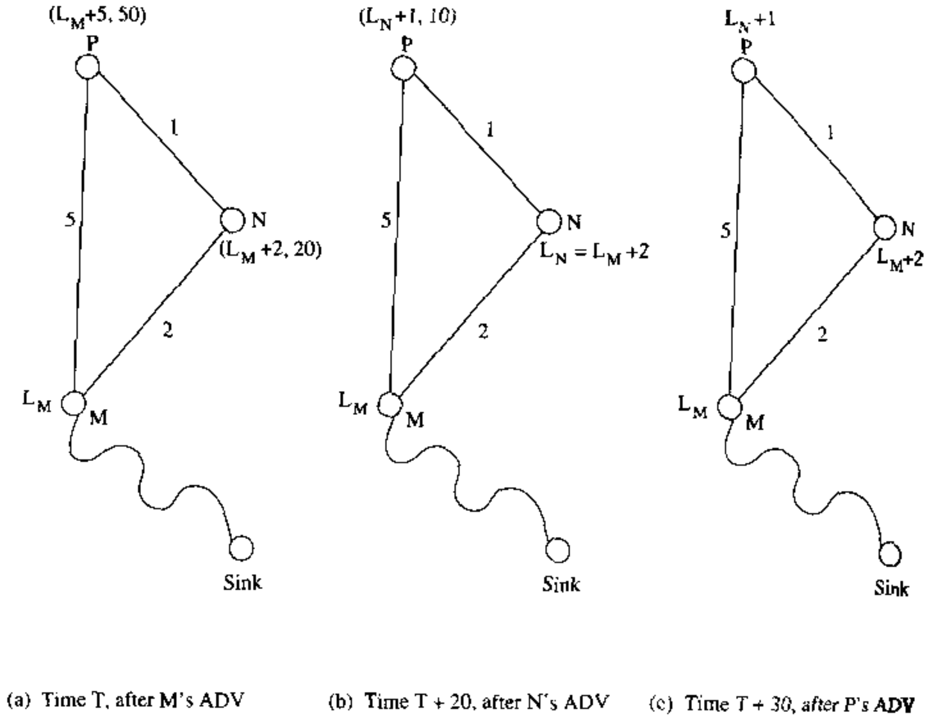

Figure 2: Setting the cost field with Cost-field approach (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

In the figure, the numbers on the links indicate the cost of the link, and the back-off timer parameter γ is set to 10. It is assumed that nodes *N* and *P* do not have a path to the sink and thus their costs to the sink are initially infinitely large. In the first step, node *M* broadcasts ADV message that is received by nodes *N* and *P*. They update their costs to $L_M + 2$ and $L_M + 5$, respectively, they also set back-off timers to 20 and 50, respectively. The second step shows a situation after 20 time units. Now the back-off timer of node *N* has expired, thus node *N* broadcasts cost $L_M + 2$ forward using an ADV message. Node *P* hears this message, and since $L_N + 1 = L_M + 3 < L_M + 5$, node *P* updates its costs and sets a back-off timer to 10. This example highlights the importance of back-off timers, node *P* never advertises its previous, non-optimal, cost of $L_M + 5$ because back-off timers are used. In the final stage, 30 time units have passed since the beginning, now the new back-off timer of node *P* has expired, node *P* sends its cost $L_N + 1$ forward with an ADV message.

The second phase of cost-field approach is data dissemination using costs. After the cost field has been established, the source sends its message to the sink *S* with a cost C_S, the message also contains cost-so-far field that is set to 0. Each node between the source and the sink will forward this message if the cost-so-far field plus its own cost equals the original source-to-sink cost, nodes also update cost-so-far field while forwarding. Such a method ensures that the optimal path is chosen between the source and the sink.

Cost-field approach is much more efficient method compared to flooding since among neighbors of a node, only neighbor that resides on the optimal path will forward the message. However, there exist some disadvantages. Setting up the cost field produces some overhead, and using back-off timers will slow down this set up process since nodes will wait some time before advertising their costs to other nodes.

3. Data gathering

The aim of data gathering is to transmit data that has been collected by the sensor nodes to the base station. Data gathering algorithms aim to maximize the amount of rounds of communication between nodes and the base station, one round means that the base station has collected data from all sensor nodes. Thus, data gathering algorithms try to minimize power consumption and delay of the gathering process.

Data gathering may seem similar to data dissemination, but there are some differences. In data dissemination, also other nodes beside the base station can request the data while in data gathering all data is transmitted to the base station. In addition, in data gathering data can be transmitted periodically, while in data dissemination data is always transmitted on demand.

Various data gathering approaches like direct transmission, PEGASIS [4], and binary scheme [5] will be covered here in more detail.

3.1 Direct transmission

In the direct transmission method all sensor nodes send their data directly to the base station. While direct transmission is a simple method, it is also very inefficient. Some sensor nodes may be very far away from the base station, thus the amount of energy consumed can be extremely high. In addition, sensor nodes must take turns when transmitting data to the base station to avoid collision. Thus, the delay is also very high. Overall, direct transmission method performs very poorly since the aim of data gathering approaches is to minimize both the energy consumption and the delay.

3.2 PEGASIS

Power-Efficient Gathering for Sensor Information Systems (PEGASIS) is a data gathering protocol that assumes that all sensor nodes know the topology of the whole network. PEGASIS aims to minimize the transmission distances over the whole sensor network, minimize the broadcast overhead, minimize the number of messages that are sent to the base station, and to distribute the energy consumption equally between all nodes.

In PEGASIS a chain of sensor nodes is constructed using a greedy algorithm starting from the node farthest from the base station. This chain is constructed before the data transmission begins and is reconstructed if nodes die out. During the data transmission, nodes aggregate the data and only one message is forwarded to the next node. The node that is selected as a leader then transmits all the data to the base station in a single message. The delay of messages reaching the base station is $O(N)$ where N is the number of sensor nodes in the network. An example of PEGASIS is shown in Figure 3 below. Data is transmitted from both ends of the chain to the leader, which sends all data to the base station.

PEGASIS achieves its goals: transmission distances over the whole network are short, overhead is relatively small, only one message is sent to the base station and energy is distributed quite equally between all nodes, since almost all nodes will send and receive exactly one message. Disadvantages of PEGASIS include high delay, in large sensor networks the chain becomes very long and a large

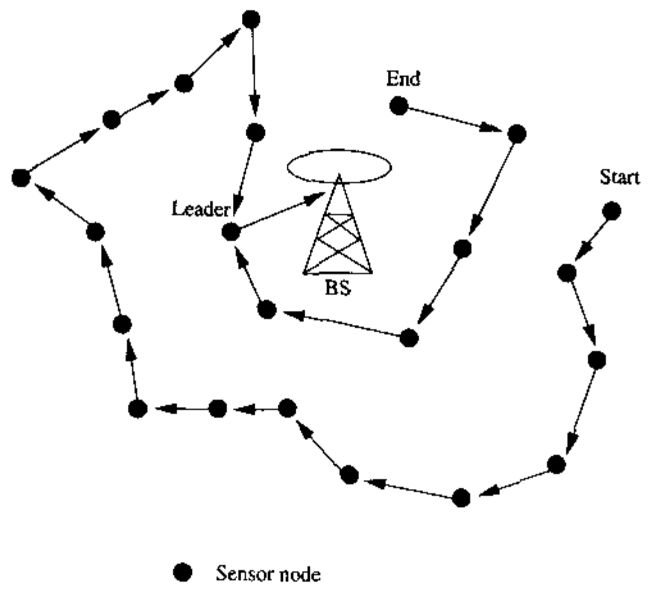

Figure 3. Data gathering with PEGASIS (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

number of hops is required to forward data from the end points of the chain to the base station. In addition, PEGASIS assumes that every node has topology information about the network and this assumption is not always valid in sensor networks.

3.3 Binary scheme

The binary scheme is also a chain-based scheme like PEGASIS. It classifies nodes into different levels. All nodes that receive message at one level rise to the next level where the amount of nodes is halved. Transmission on a one level occur simultaneously to reduce delay. An example of the binary scheme is shown in Figure 4 below. Nodes s1, s3, s5 and s7 receive messages on the first level and thus they rise to the next level. On the second level nodes s3 and s7 receive messages and finally node s7 forwards all data to the base station.

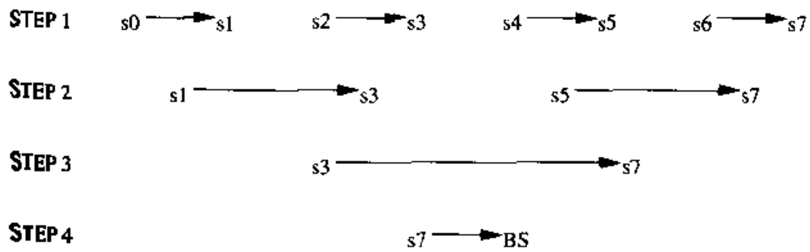

Figure 4. Binary scheme (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

Biggest advantage of binary scheme is a very low delay of only *O(log₂N)*, where the *N* is the amount of nodes. Thus, binary scheme has significantly lower delay than PEGASIS in large sensor networks. However, binary scheme relies on simultaneous transmission which are possible if the nodes communicate using CDMA, but the scheme does not work with all networks. Other disadvantages include non equal distribution of energy consumption, nodes that are active on several levels consume more energy than nodes that are only active at the first level. This might lead to the situation where some of sensor nodes die earlier than others. In addition, transmission distances may become long in high levels, which leads to a high power consumption.

For other networks, similar chain-based three-level scheme [5] has been developed. It divides the chain in groups and within a group only one node is transmits at once. Then the leader of the group rises to the next level where the first level leaders transmit data to a new leader, like in the binary scheme. At the third level, all data is transmitted to a single node that passes it to the base station. Three levels have been found to optimize delay and power consumption.

4. Conclusions

There exist several different approaches for data dissemination and gathering. Simple approaches like flooding and direct transmission are very ineffective. More complex approaches are more effective, but they also have disadvantages and they may not work with all sensor networks. Best method depends on the situation: are there a large amount of queries in data dissemination situation? How many sensor nodes there are overall? Do all sensor nodes know topology of the whole network?

References

[1] C. Murthy and B. S. Manoj. Ad hoc Wireless Networks: Architectures and Protocols. 2004.
[2] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Network. In Proceedings of ACM MOBICOM 1999, pp. 174-185, August 1999.
[3] F. Ye, A. Chen, S. Lu, and L. Zhang. A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks. In Proceedings of IEEE ICCCN 2001, pp. 304-309, October 2001.
[4] S. Lindsey and C. S. Raghavendra. "PEGASIS: Power-Efficient Gathering in Sensor Information System". In Proceedings of IEEE ICC 2001, vol. 3, pp. 1125-1130, June 2001.
[5] S. Lindsey, C. S. Raghavendra, and K. M. Sivalingam. Data-Gathering Algorithms in Sensor Networks Using Energy Metrics. IEEE Transactions on Parallel and Dsitributed Systems, vol. 13, no. 9, pp. 924-935, September 2002.