Query Dissemination with Predictable Reachability and Energy Usage in Sensor Networks

Zinaida Benenson[1,⋆], Markus Bestehorn[2], Erik Buchmann[2], Felix C. Freiling[1], and Marek Jawurek[3]

[1] University of Mannheim, Germany
[2] University of Karlsruhe, Germany
[3] Fraunhofer IESE, Germany

Abstract. Energy-efficient query dissemination plays an important role for the lifetime of sensor networks. In this work, we consider probabilistic flooding for query dissemination and develop an analytical framework which enables the base station to predict the energy consumed and the nodes reached according to the rebroadcast probability. Furthermore, we devise a topology discovery protocol that collects the structural information required for the framework. Our analysis shows that the energy savings exceed the energy spent to obtain the required information after a small number of query disseminations in realistic settings. We verified our results both with simulations and experiments using the SUN Spot nodes.

1 Introduction

Wireless sensor networks have been established in many important application areas from ambient intelligence over scientific research to industrial uses. Such sensor networks usually consist of numerous battery-powered nodes [3,2] equipped with sensing devices, low-power wireless communication and limited computational resources. In order to fulfill complex measurement tasks, the sensor-nodes use self-organization techniques to form ad-hoc networks where the nodes (1) forward queries from a central base station, (2) measure sensor values, (3) do in-network query processing and (4) return the results to the base station. In this paper, we focus on the query dissemination phase, i.e., the first step of query processing in sensor networks.

One of the most important optimization goals in sensor networks is to maximize their lifetime by minimizing the energy spent for communication. However, saving communication effort obviously may have a negative effect on quality-of-service parameters of the query. For example, if energy is saved by querying only 50% of the nodes, the accuracy of the query degrades. How much it degrades depends on many factors and is not very well understood. Quantifying this tradeoff

⋆ Zinaida Benenson was supported by Landesstiftung Baden Württemberg as part of Project "ZeuS" and by the Schlieben-Lange scholarship.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 279–292, 2008.

between communication strategy and service quality for query dissemination is the topic of this paper.

Related Work. While numerous sophisticated in-network query processing techniques have been developed [11,12,18,19], they mostly focus on operator processing, optimization and aggregation techniques. The dissemination of the query from the base station into the network has either been disregarded or is done via simple flooding [8,14]. It is well known that flooding wastes energy. For example, analyses [13] have shown that a rebroadcast increases the area where the message is received by 61% at most, dropping to $\approx$ 20% for average networks. Therefore, most of the rebroadcasts will not result in additional nodes receiving the query. Furthermore, most nodes receive the query more than once, which results in additional energy consumption because receiving messages also consumes energy.

To avoid the disadvantages of simple flooding, several mechanisms for broadcasting in wireless networks have been proposed (see [17] for an overview). Generally, these approaches try to control which nodes rebroadcast a message in order to keep the number of nodes that receive the query more than once as small as possible. For example, in *counter-based flooding* schemes [13,17], if a node hears k or more of its neighbors rebroadcast the message, it suppresses its own transmission. In *neighbor knowledge* broadcasting schemes [10,15], nodes use local topology information to determine which nodes must rebroadcast a message. The advantage of these approaches is that the overlap of recipients can be reduced in a controlled manner, but this comes at the significant cost of acquiring and updating the neighborhood information. Furthermore, [16] has shown that finding a minimal set of rebroadcasting nodes can be reduced to the Dominating Set Problem, which is NP-complete [6].

A very promising approach are *probabilistic* or *epidemic* broadcast algorithms [13,5] where every node forwards a message with a predefined probability p. Compared to schemes using neighborhood knowledge, these methods do not induce the overhead of acquiring, storing and updating neighborhood knowledge. However, these schemes require information about the network in order to determine an optimal p. If p is set too high, the disadvantages of simple flooding arise, and if p is too low, the probability that all nodes receive the broadcast message decreases. In this paper we will focus on probabilistic flooding.

Contributions. In this paper, we study query dissemination techniques that can be seen as a combination between neighbor knowledge broadcasting and probabilistic flooding. Using extensive simulations we explore the tradeoff between energy, reachability and structural information required. We show that using very moderate structural information on the network it is possible to predict the number of nodes reached according to a certain broadcast probability p. Furthermore, the number of transmissions can be estimated in advance.

In particular, we make the following contributions:

1. We introduce an analytical framework to estimate the reachability and the number of transmissions in dependence to the rebroadcast probability p. Our

framework bases on connectivity information and a histogram containing the number of nodes reached with each rebroadcast, starting at the base station.
2. We describe a lightweight distributed topology discovery protocol which obtains the required information. Our analysis shows that gathering structural information and computing an optimal p saves energy after a small number of probabilistic floodings in realistic settings.
3. We conducted simulations with up to 425 nodes to verify the results of our framework for large numbers of nodes. Furthermore, we tested our findings on a testbed consisting of 17 Sun Spot sensor nodes.

Outline. In Section 2 we present a framework which estimates the number of nodes reached and energy spent by probabilistic flooding for a particular rebroadcast probability p. The framework depends on topological information. In Section 3 we show how to gather the required information efficiently. In Section 4 we present simulation and experimental results, and we conclude in Section 5.

2 Reachability and Energy Consumption Prediction for Query Dissemination

In this work we focus on probabilistic flooding where each node rebroadcasts queries with a fixed probability p. Parameter p allows to fine-tune the tradeoff between energy spent for query dissemination and the number of nodes reached. Moreover, in most (densely connected) sensor networks there exists a $p_0 < 1$ such that all nodes are reached by the base station. Thus, if the rebroadcast probability p is larger than p_0, more queries are rebroadcast than necessary, and the query dissemination can save energy by using p_0. On the other hand, if $p < p_0$, the query reaches only a fraction of nodes. This can be useful to trade energy with result quality.

Our goal is to develop a framework to predict for every p the number of reached nodes R and the energy E consumed by the query dissemination process. Knowing the dependencies between p, R and E allows the base station to estimate how many nodes can be reached using a fixed amount of energy, or at which p the reachability cannot be improved any more (at least, for reasonable energy cost). Obviously, energy usage prediction depends on reachability prediction, which in turn depends on the network topology. The more the base station knows about network topology, the more precise prediction can be made. On the other hand, gathering information about network topology consumes energy. Thus, we are interested in making predictions using a set of topological information which can be obtained without exhausting potential energy savings due to deriving an optimal p.

In the following we will present our framework for predicting reachability $R(p)$ and energy consumption $E(p)$ according to given topological information and a rebroadcast probability p. More specifically, $R(p)$ estimates the number of nodes reached, and $E(p)$ provides an estimate for the number of sent and received messages, which is proportional to the energy consumed.

2.1 Assumptions and Notations

Our estimation of the reachability bases on two assumptions:

- The sensor network is in a stable state while flooding the query, i.e., the number of nodes in each hop set does not change significantly between obtaining topology information and flooding.
- A node is either reached by a node that is one hop closer to the base station, or has the same hop distance to the base station.

A flooding disperses through a topology in multiple steps, beginning at the base station. The nodes which receive the query directly from the base station (1 hop) rebroadcast it, so that the query reaches the nodes two hops away from the base station in the next step. The procedure recurs until each node has forwarded the message once.

If a node A receives a previously unknown flooding message from a node B, we say that A is reached by B in this particular *flooding instance*. In addition, we will denote all nodes reached with h hops as *hop set* $H[h]$.

2.2 Topological Information

Our analytical framework depends on the following topological information (Section 3 will introduce a protocol that collects it efficiently):

- $histogram[h]$: stores the number of nodes reached at each hop from the base station, i.e., $\forall i \in \{1 \cdots n\} : histogram[i] = |H[i]|$.
- $connectivity[h]$ stores the average number of connections from one node in hop set $H[h]$ to a node from $H[h-1]$.
- $interconnectivity[h]$ stores the average number of connections between the nodes from the same hop set, i.e., the connections a node in $H[h]$ has to another node in $H[h]$.

Figure 1 illustrates this with an example. In this figure the hop set $H[i]$ consists of 3 nodes, the previous hop set $H[i-1]$ consists of 2 nodes. Edges connect the nodes that can hear each other's broadcast. Figure 2 shows the histogram and (inter-)connectivity for the example in Figure 1.

2.3 Reachability Prediction

Let $R_{direct}(h,p)$ be the number of nodes in hop set h which received their flooding message directly from a node in the hop set $H[h-1]$, and let $R_{indirect}(h,p)$ denote the number of nodes which received the flooded message from a node in the same hop set $H[h]$. Then the number of nodes reached at the h-th hop for a specific rebroadcast probability p can be computed as follows:

$$R(h,p) = min(R_{direct}(h,p) + R_{indirect}(h,p), histogram[h]) \tag{1}$$

The total reachability for some p is the sum over all hops:

$$R(p) = \sum^{h} R(h,p) \tag{2}$$

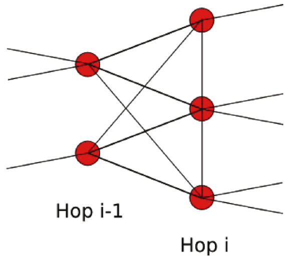


Fig. 1. Example for hop sets and their (Inter-)Connectivity

...	$i-1$	i	...
...	2	3	...

(a) Histogram

...	$i-1$	i	...
...	1.5	2	...

(b) Connectivity

...	$i-1$	i	...
...	0	4/3	...

(c) Interconnectivity

Fig. 2. Histogram, Connectivity and Interconnectivity in Figure 1

Note that $R_{direct}(h,p) + R_{indirect}(h,p)$ can be larger than the actual number of nodes in the hop set $H[h]$, because rebroadcast messages can be received from nodes which might have received the message before. Thus the minimum function ensures that at most the actual number of nodes in the hop set is returned. $R_{direct}(h,p)$ can be computed recursively: $histogram[h-1]$ nodes could forward a message directly to a node in $H[h]$, but only $k = p * R(h-1,p)$ of $histogram[h-1]$ nodes have received the message in the previous step.

Let $P(\text{event})$ denote the probability for a certain event. Now we need the probability for the event "A node from hop set $H[h]$ receives its message from a node from hop set $H[h-1]$" The probability for this event is:

$$P(\text{reached directly}) = 1 - P(\text{not reached directly}) \tag{3}$$

The counter-event "not reached directly" can be obtained by considering the nodes which did not receive the message in the previous step. Thus, the problem corresponds to an urn model where k black and $n-k$ red balls are placed in an urn, and $P(\text{not reached directly})$ means to draw red balls only. Let $l = connectivity[h]$ be the number of connections a node in $H[h]$ has to the previous hop set $H[h-1]$ on average. The probability $P(\text{not reached directly})$ can be computed as follows:

$$P(\text{not reached directly}) = \prod_{l=0}^{\lceil connectivity[h] \rceil - 1} \frac{n-l-k}{n-l} \tag{4}$$

After having obtained this probability, we can calculate the number of nodes from hop set $H[h]$ receiving the flooding directly by multiplying the probability for the opposite case with the number of nodes in the hop set:

$$R_{direct}(h,p) = P(\text{reached directly}) * histogram[h] \tag{5}$$

The remaining nodes in hop set $H[h]$ can still be reached indirectly, i.e., by a subsequent broadcast by nodes from the same hop set. To calculate the number of nodes reached indirectly, we assume that the nodes which received the message are equally distributed over the hop set, i.e., if k from n nodes are directly reached, each node in the hop set obtained the message with probability $\frac{k}{n}$. Our experimental evaluation will show that this simplification is legitimate, i.e, it is not necessary to collect topological information in more detail. We calculate the number of neighbors of a node which directly received the flooding message and then rebroadcast it as:

$$n_{dr} = P(\text{reached directly}) * interconnectivity[h] * p \tag{6}$$

Finally, we estimate the number of nodes which received the flooding message indirectly:

$$R_{indirect}(h,p) = n_{dr} * histogram[h]. \tag{7}$$

2.4 Energy Consumption Prediction

After having estimated the number of nodes reached, we will estimate the energy required by probabilistic flooding. Therefore, we distinguish between sent and received messages. The number of messages sent in hop set $H[h]$ is as follows:

$$msgs_{sent}(h,p) = R(h,p) * p \tag{8}$$

Next, we estimate the number of messages received from the nodes of the previous hop:

$$Rec_1(h,p) = R(h-1,p) * p * \frac{connectivity[h] * histogram[h]}{histogram[h-1]} \tag{9}$$

$\frac{connectivity[h]*histogram[h]}{histogram[h-1]}$ calculates the average number of outgoing links from hop set $H[h-1]$ to $H[h]$. The number of all "receive" events induced at nodes of the hop set $H[h]$ and hop set $H[h-1]$ by the rebroadcast of reached nodes of hop set $H[h]$ can be calculated as follows:

$$Rec_2(h,p) = R(h,p) * p * (connectivity[h] + interconnectivity[h]) \tag{10}$$

Finally, the total number of received messages can be estimated as

$$msgs_{received}(h,p) = Rec_1(h,p) + Rec_2(h,p) \tag{11}$$

The total energy cost of the probabilistic flooding is calculated by vector multiplication of the tuple of sent and received messages with the vector of energy costs for sending and receiving and adding them up for every hop set:

$$E(p) = \sum_h (msgs_{sent}, msgs_{received})_{(h,p)} * \begin{pmatrix} energyPerSend \\ energyPerReceive \end{pmatrix} \tag{12}$$

3 Topology Discovery Protocol

We now describe the light-weight topology discovery protocol used in our experiments. It is an adaption of the well-known *echo algorithm* by Chang [4], i.e., it is structured in two waves: The first *expansion* wave of messages is flooded from the base station and is used to explore the network. When this waves reaches the borders of the network, a second *contraction* wave flows back to the base station, aggregating topology information / histograms on its way. The prediction formulas presented in Section 2 use these histograms to determine the parameter p for probabilistic flooding. Due to space limitations, we only present the general idea of the protocol here.

The base station initiates the topology discovery by broadcasting a *Topology Discovery Message* (`TDReq`), thus starting the expansion wave.

Expansion Wave. When a node receives a `TDReq` for the first time, the receiver must accomplish 4 steps:

1. Create an empty histogram data structure as described in Section 2.2 and mark the sender of the `TDReq` as its parent node. The receiver also extracts the hop number from the `TDReq` and stores it.
2. Start a timeout to ensure that the receiver does not wait forever for potential children.
3. Broadcast own request message with the receiver as sender, an incremented hop number, and parent id of the receiver.
4. Wait until the afore mentioned timeout expires. Note that the timeout should be sufficiently long to allow the children of the node to receive, process and rebroadcast their own `TDReq` messages. When the timeout expires, the contraction phase starts.

If a `TDReq` is received, then it could have three different originators. It could either be (1) a sibling of the node's parent, (2) a sibling of the node itself, or (3) a node in the subsequent hop set. Note that all three cases can be distinguished from the information contained in the `TDReq`. For example, in case (3) the request will contain the id of the receiver node. Depending on the case, the connectivity or inter-connectivity value in the histogram data structure is modified.

Contraction Wave. While a node waits for the timeout to expire, all incoming *Topology Discovery Responses* are recorded into the histogram data structure. On leaf nodes, the timeout expires without any incoming response messages, thus leaf nodes create response messages containing their hop number and appropriate values for connectivity and inter-connectivity[1]. Every leaf node sends such a response message to its parent and thereby starts the contraction wave.

[1] The average values for connectivity / inter-connectivity are stored as tuples to allow aggregation: The first value contains the sum of connections and the second stands for the number of nodes which have aggregated these connections. This allows the aggregation at every node and avoids floating point numbers in messages.

In case the node has children, the histogram lists of every response are aggregated in a way that the position i of the resulting list contains the sum of the histograms of the children. These aggregates histograms are always forwarded to the parent node and eventually reach the base station. Based on these values, the base station is able to predict reachability and energy consumption as described in Section 2.

Energy Cost and Message Size of Topology Discovery Protocol. As every node only broadcasts one Topology Discovery Request and only sends one Topology Discovery Response, the energy costs per node can be estimated as follows:

$$E_{Node} = E_{send}(b_1)+E_{send}(b_2)+AverageNodeDegree*(E_{rcv}(b_1)+E_{rcv}(b_2)) \quad (13)$$

The value b_1 stands for the number of bytes in the Topology Discovery Request of the node, b_2 stands for the number of bytes in the Topology Discovery Response.

Later we calculate energy consumption of Topology Discovery Protocol for a particular scenario and show after how many probabilistically flooded queries the protocol pays off.

4 Evaluation

In this section we evaluate the prediction framework with different node setups using simulations and a deployment of 17 Sun SPOT sensor nodes [2] in our faculty building. We compare the predictions made by our framework with the flooding of queries in simulated networks of up to 425 nodes and in the real sensor network, showing the following:

1. For all simulated networks and the real sensor network, the accuracy of the reachability prediction based on the topology information is sufficiently high.
2. Any inaccuracy related to the probabilistic flooding is clearly outweighed by the amount of energy saved through decreased communication overhead.

Our framework produces stochastic results for the average case, i.e., it works well for sufficiently dense networks or for large numbers of trials.Thus, we expect a deviation between the predicted values and experimental results. Nevertheless, our predictions can be successfully used for query optimization purposes, lifetime estimation or the computation of the rebroadcast probability with a small additional safety margin.

4.1 Simulations

For the simulation we used a custom *Karlsruhe Sensor Networking Simulator* which is interface-compatible to Sun SPOT sensor nodes, thus enabling us to deploy the prediction framework as well as the topology discovery protocol in both the simulated environment and the real deployment.

Simulations Setup. We considered the following simulation scenarios: uniform and Gaussian distributed nodes, a scale-free distributed scenario, and a real world set up from the Intel Lab Website [1]. Due to space limit we only present the results for the first two scenarios below.

Uniform node distribution. All topologies of this scenario distribute the sensor nodes uniformly in a circular area around the centre where the base station is located. The parameter of this scenario is the average number of neighbors of every node. The radius of the simulation area is fixed, and the number of nodes is adjusted accordingly to obtain the respective average node degree. We used networks of node degrees 4, 8, 12 and 16, and generated for each node degree 40 different topologies. For each topology we ran 100 experiments.

Table 1. Average node degree in Uniform scenario and resulting amount of nodes

Average Node Degree	Used Sensors
4	125
8	225
12	325
16	425

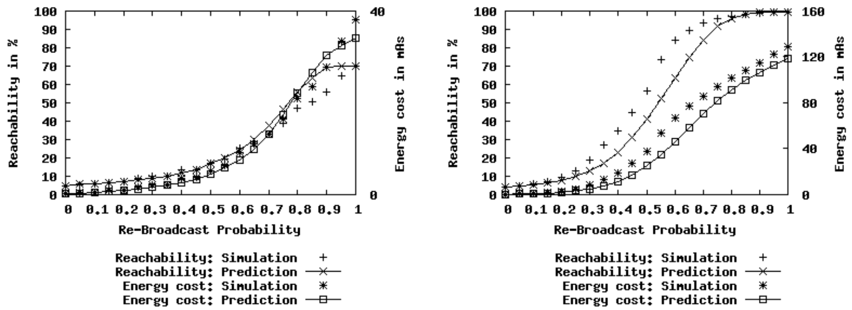
Gaussian node distribution. In this scenario all sensor nodes are distributed using a Gaussian distribution over an area with a fixed radius of 30 units. The coordinates of the nodes are taken from a Gaussian sampling with the centre of the environment as mean and a standard deviation of 18 units. By choosing this standard deviation most of the sensors are placed in the target area, only few nodes were placed beyond. Most nodes are located close to the centre, and the further away from the base station the lower the node density. This scenario has the number of sensor nodes placed as parameter. In order to compare the results from the uniform scenario with this scenario, we generated instances with the same average node degrees for scenarios with the same number of nodes (see Table 1). As in uniform scenario, for each of the four network sizes we generated 40 topologies and run 100 experiments per topology.

Reachability and Energy Consumption. For this series of experiments, we assume a message payload size of 28 bytes for the query. According to an analysis [9] of MICAz [3] sensor nodes the energy consumption Formulae 14 and 15 were determined. Parameter b specifies the number of bytes sent/received.

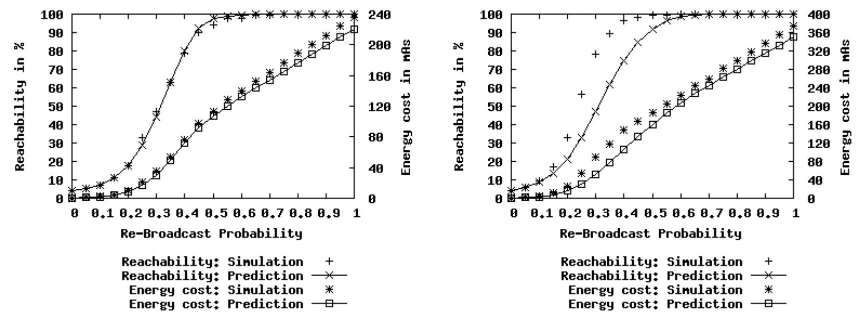
$$EnergyForSending(b) = 0.185191mAs + (b - 28byte) * 2.48461mAs * 10^{-5} \quad (14)$$

$$EnergyForReceiving(b) = 0.042mAs + (b - 28byte) * 2.47915mAs * 10^{-5} \quad (15)$$

The energy consumption was firstly measured for standard TinyOS [7] message payload of 28 bytes, and then the energy consumption for sending (receiving) b additional bytes was determined. The results of evaluation of our



(a) average node degree 4 (125 nodes) (b) average node degree 8 (225 nodes)



(c) average node degree 12 (325 nodes) (d) average node degree 16 (425 nodes)

Fig. 3. Comparison of simulated reachability/energy cost in uniform scenarios

reachability and energy consumption prediction framework are presented in Figure 3 for the uniform scenario, and in Figure 4 for the Gaussian scenario.

One can see that our framework works reasonably well in sufficiently dense scenarios. It systematically underestimates reachability and energy consumption, but it still allows to save a large amount of energy. For example, in Figure 3(b–d), although the full reachability is achieved with smaller rebroadcast probabilities than predicted, flooding with the predicted probability still allows to save from 10 (b) to 37 (d) percent of energy. Moreover, reachability and energy consumption predictions for the Gaussian scenario follow the simulated results so closely that they allow very accurate determination of the rebroadcast probability needed to reach a particular amount of nodes. Note that in Gaussian scenarios, some nodes are placed so far from the base station that the network becomes disconnected.

Topology Discovery and Reachability Prediction Payoff. Assuming a uniform scenario with 425 nodes, average node degree 16 and a reachability of about 99%, up to 150 mAs can be saved using our prediction framework (see Figure 3(d)). Using rebroadcast probability $p = 0.6$ only approximately $220mAs$ are consumed in comparison to the simple flooding which consumes

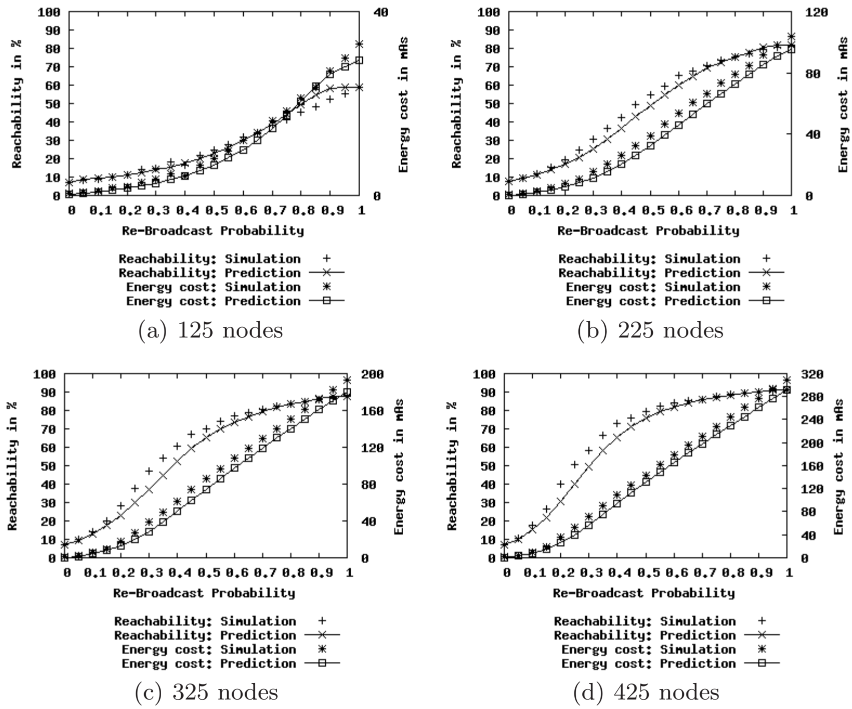


Fig. 4. Comparison of simulated reachability/energy cost in Gauss scenarios

$370mAs$. However, some energy was previously spent for Topology Discovery Protocol. Using Formula 13 for energy consumption of the Topology Discovery Protocol, and Formulas 14 and 15 for energy consumption of MICAz nodes, we estimated that in the above scenario, the Topology Discovery Protocol has approximate costs of $722mAs$ (we omit the computations due to space limit). Thus, the Topology Discovery Protocol would have paid off after 5 probabilistic query floodings.

4.2 Sun SPOT Deployment

After having provided simulation results, we tested our framework together with the topology discovery protocol in real testbed. Figure 5 shows a map of 17 Sun SPOT sensor nodes (circles) and a base station (square) that are deployed in the offices at the Institute for Programming Structures and Data Organization (IPD) of the University of Karlsruhe. On each node we counted incoming and outgoing messages, as well as the sizes of the messages in bytes. These values were stored in the memory of each node and collected after the experiments were finished.

To assess the quality of the flooding prediction, the following experiment was repeated 10 times:

1. A simple flooding of a query was executed to determine the number of reached nodes for simple flooding.
2. Using the topology discovery protocol, the information required for the prediction was collected.
3. Using the topology information, the parameter p for the probabilistic flooding was computed with the aim of disseminating a query to all nodes of the network. Thus, we tried to determine the lowest p for which a reachability of 100% was predicted.
4. Based on the computed value of p, a query message was flooded into the network using probabilistic flooding.

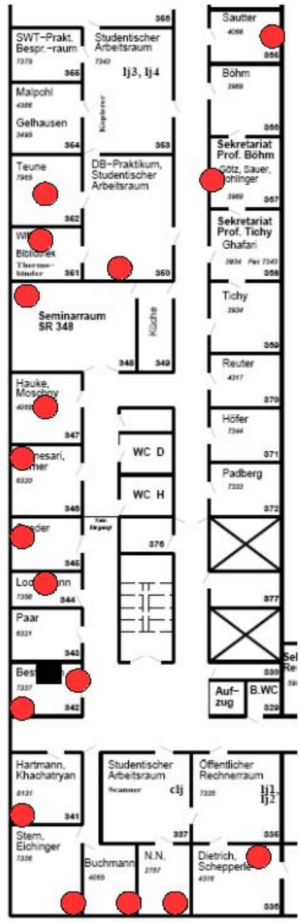


Fig. 5. Map of 17 Sun SPOTs and a Base Station deployed at the IPD

Despite minor changes between the different experiments within the topology information, which can be attributed to environmental influences (e.g. open/closed doors in the used offices), the topology information was consistent throughout our experiments.

Table 2 shows the average results for the 10 experiments: Generally, the accuracy of the prediction is sufficient, even though there is a small difference between the 16.3 nodes reached by simple flooding compared to the probabilistic flooding with 15.4 nodes reached on average.

Table 2 shows messages required by the simple and the probabilistic flooding: The number of messages sent and received when the probabilistic flooding is used, is by far lower than the amount used by the simple flooding. Thus the amount of saved energy due to reduced communication clearly outweighs the small inaccuracy of the prediction.

Table 2. Result of the flooding experiment using the Sun SPOT deployment

Flooding	Avg. Reached Nodes (of 17)	Messages Sent	Messages Received
Simple	16.3	16.3	63.8
Probabilistic	15.4	10.2	34

5 Conclusions and Future Work

It is challenging to realize energy-efficient query dissemination with predictable reachability and energy usage in sensor networks: Unnecessary transmissions should be generally avoided in order to save energy. On the other hand, it requires knowledge about the sensor network to find out which transmissions are actually required, but obtaining these information comes with an additional communication overhead.

In this paper we have used probabilistic flooding as a model to explore the relations between (1) energy consumption of the query dissemination phase, (2) the number of nodes reached and (3) the energy spent to gather structural information about the network which are required to parameterize probabilistic flooding. In particular, we have introduced an analytical framework that enables the base station to estimate the reachability and energy consumption of probabilistic flooding according to based on connectivity information. Furthermore, we have shown how to gather such information efficiently, and we have computed the break-even between energy saved and energy spent to obtain structural information. Both experiments with a simulator and an implementation with a testbed consisting of 17 SUN Spot nodes validate our findings.

As part of our future work we plan to consider "back links" in flooding, and other query dissemination strategies. In addition, we are interested in the relations between the energy spent for query dissemination and the accuracy of the query result returned.

References

1. Intel berkeley research lab data, `http://db.csail.mit.edu/labdata/labdata.html`
2. SUN Microsystems Inc., Small Programmable Object Technology (SPOT)
3. Xbow technology inc. wireless sensor networks
4. Chang, E.J.H.: Echo algorithms: Depth parallel operations on general graphs. IEEE Transactions on Software Engineering 8(4), 391–401 (1982)
5. Eugster, P.T., Guerraoui, R., Kermarrec, A.-M., Massoulieacute, L.: Epidemic information dissemination in distributed systems. Computer 37(5), 60–67 (2004)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness (1990)
7. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.S.J.: System architecture directions for networked sensors. In: Proc. 9th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (2000)
8. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw. (2003)
9. Kellner, S., Pink, M., Meier, D., Blaß, E.-O.: Towards a realistic energy model for wireless sensor networks. In: WONS 2008 (to appear) (January 2008)
10. Lim, H., Kim, C.: Multicast tree construction and flooding in wireless ad hoc networks. In: MSWIM 2000: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (2000)
11. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. In: SIGOPS (2002)

12. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tinydb: an acquisitional query processing system for sensor networks. In: ACM TODS (2005)
13. Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network. In: MobiCom 1999: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (1999)
14. Obraczka, K., Viswanath, K., Tsudik, G.: Flooding for reliable multicast in multi-hop ad hoc networks (2001)
15. Peng, W., Lu, X.-C.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: MobiHoc 2000: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing (2000)
16. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: HICSS 2002: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), vol. 9 (2002)
17. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC) (2002)
18. Yao, Y., Gehrke, J.: The cougar approach to in-network query processing in sensor networks. In: SIGMOD Rec. (2002)
19. Yao, Y., Gehrke, J.: Query processing in sensor networks. 2003. In: CIDR 2003: Proceedings of the First Biennial Conference on Innovative Data Systems Research (2003)