The Heathland Experiment: Results And Experiences

V. Turau, C. Renner, M. Venzke, S. Waschik, C. Weyer, and M. Witt

Hamburg University of Technology, Department of Telematics
Schwarzenbergstraße 95, 21073 Hamburg, Germany

turau@tu-harburg.de

ABSTRACT

This paper reports on the experience gained during a real-world deployment of a sensor network based on the ESB platform in the heathlands of Northern Germany. The goal of the experiment was to gain a deeper insight into the problems of real deployments as opposed to simulated networks. The focus of this report is on the quality of radio links and the influence of the link quality on multi-hop routing.

1. INTRODUCTION

In recent years wireless sensor networks have been attracting research interest given the recent advances in miniaturization and low-cost, low-power design. Many algorithms have been proposed to solve the problems inherent to sensor networks, foremost resource limitations and high failure rates. The vast majority of algorithms has not been implemented on real sensor networks, but evaluated using simulation tools. Simulations are a valuable and cheap means to compare specific aspects of different algorithms solving the same problem (e. g., routing or data aggregation), but currently no simulation tool is capable to allow for all imponderabilities of a real deployment of a sensor network in a harsh environment over a longer period of time. To attain a deeper insight into sensor networks, experiments with real deployments are indispensable. But up to today the number of deployed wireless sensor networks is extremely low compared with the number of publications.

Environmental monitoring is a significant driver for wireless sensor network research, promising dynamic, real-time data about monitored variables of an area and so enabling many new applications. Because of this, it comes as no surprise, that almost all real experiments were conducted with this application background. In particular, the first published experience with real deployments of sensor networks were about habitat monitoring [4]. Only recently other application backgrounds such as wildfire monitoring were considered in real experiments [1].

In this paper we report about an experiment with a real deployment of a sensor network with 24 nodes running for two weeks in March 2005 in the heathlands of Northern Germany. After the deployment the application ran without any human attention. At the time of writing this report the experiment was just finished. The preliminary results allow an insight into the problems emerging during the deployment and operation of a sensor network and provide valuable information for other installations. The focus of the following analysis is on the communication between the nodes.

2. THE GOALS

The Heathland experiment is to our knowledge the first outdoor long term usage of the Embedded Sensor Board (ESB) platform [3]. Naturally, we were interested in the overall performance of the system. Specifically, this experiment provided a basis to evaluate our neighborhood exchange and routing protocol and acted as an indicator for the feasibility of our deployment and debugging strategy. In particular the following aspects are analyzed in this paper:

- General radio performance (packet loss, packet errors)
- Distribution and constancy of transmission ranges
- Stability and quality of links as determined by the neighborhood protocol
- Communication in a multi-hop environment

To conduct this analysis, a considerable amount of data was logged by the sink node (about 6 MB per day). Whenever a packet with a sensor reading was sent to the sink, the state of the node (neighborhood list, remaining energy etc.) was included in the packet. Some data was also stored in the EEPROM of each node; this was used to analyze causes of failure of a node, e. g., when the node was no longer reachable by other nodes.

The Heathland experiment was not aimed at evaluating the long term operation of the sensor network, hence the application code was not optimized to achieve an increased expectation of life.

3. THE EXPERIMENT

3.1 The Hardware

For the experiment, the ESB nodes from the Free University Berlin were used [3]. They consist of the micro controller MSP 430 from Texas Instruments, the transceiver TR1001, which operates at 868 MHz at a data rate of 19.2 kbit/s, some sensors, and a RS232 serial interface. The radio transmit power can be tuned in software, with 0 being the minimum and 99 the maximum. Each node has 2 KB RAM and 8 KB EEPROM. The nodes were powered by three AA batteries. The sink had a permanent power supply. The power consumption of the nodes according to the specifications of the vendor varies from 8 μA in sleep mode up to 12 mA when running with all sensors. A description of the sensors of the ESB nodes can be found in [3].

3.2 The Packaging

The nodes had to be prepared for diverse weather conditions including snow, rain, and sunshine (see Figure 1). Waterproof packing was essential. Therefore the nodes were shrink-wrapped together with desiccant bags. The foils were then placed in waterproof boxes, which again were put into plastic bags. These bags were affixed on trees, windows etc. using gaffer tape. The recorded temperature during the experiment varied in a range of almost 40 °C. Figure 2 shows the developing of the temperature during the two weeks of the experiment as measured by the sensor nodes (in- and outdoor temperature). Our results indicate that the applied packaging was appropriate (no node failed due to extraneous cause). The applied packaging formed an insulation of the nodes and affected the measurements of the sensors. Since this was not tangent to our goals, it was accepted.

Figure 1: The packaging of the sensor nodes

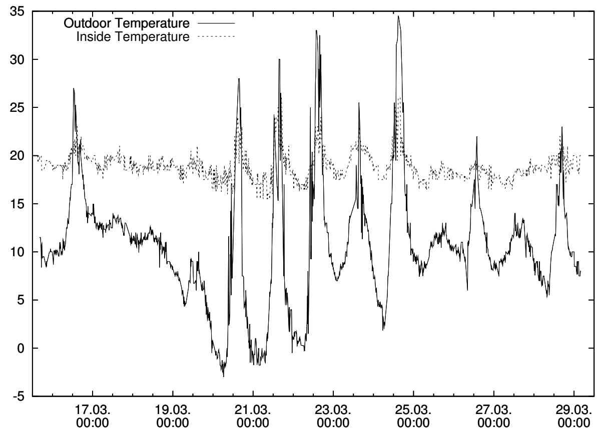

Figure 2: Recorded temperatures during experiment

3.3 The Application

A broad class of applications of wireless sensor networks are so-called *sense-and-send* applications. They share a common structure, where sensors deployed in a wide-ranging area are tasked to take periodic readings, and report results to a central repository. The implemented application follows in principle this *sense-and-send* pattern. The nodes periodically send readings from their five sensors supported by the hardware [3] to the sink. To account for topology changes caused by node failures, deployment of new nodes, decreasing communication radii due to fading battery energy, and moving nodes, routing trees and neighbor relationships were recomputed regularly and unlike other experiments (e. g. [5]) a multi-hop network was used.

The design of the software follows the style of separate layers. The bottom layer consists of a proprietary firmware that is shipped together with the ESB nodes. The included radio transmission protocol is very simple; its unreliableness is the source of many problems. The protocol silently discards a packet after 15 unsuccessful trials of submission. On top of this a neighbor discovery protocol called *Wireless Neighborhood Exploration* (WNX) has been implemented. It is a slightly modified implementation of TND, the proactive neighbor discovery protocol of TBRPF as defined in RFC 3684 [2]. WNX determines uni- and bidirectional links and adds a quality descriptor to every unidirectional link. A link becomes a bidirectional link if the qualities of both link directions exceed a given limit. Based on the bidirectional links provided by WNX, a depth-first spanning tree rooted at the sink is built. The depth-first search is performed using a distributed algorithm developed by Tsin [6]. Each node keeps a list of at most eight neighbors, high-quality links are preferred over low-quality ones. The spanning tree is used to route messages from any node to the sink. The reason for using depth-first spanning trees is twofold:

- In order to better test multi-hop communication, routing trees with a higher depth are needed. Depth-first trees usually have a higher depth than breadth-first trees, which can easily be computed using flooding.
- Since depth-first search is the basis for many algorithms which are likely to be employed in a sensor network, we were interested in the performance of our distributed implementation.

The neighborhood relation is used to build the routing tree. Therefore, WNX is only executed at the beginning of every application cycle. For this reason a suspend mode was added to WNX. Since the leaves of a routing tree are not needed to route messages to the sink, they only turn on their radio:

- while running WNX,
- during the depth-first search, and
- when they send their sensor readings to the sink.

As a consequence, leaf nodes turn off their radio for about 46 minutes during every hour.

After the completion of the depth-first search all nodes send their neighbor list including the quality factors to the sink. After this step all leaf nodes send their data in intervals of 10 minutes to the sink. To reduce the likelihood of interferences, the times a leaf node sends its readings are randomly distributed in this interval. Apart from the sensor readings the following data is included in every packet:

- time-stamp with respect to local clock
- remaining battery energy
- number of packets received and sent
- number of packet transmission retries
- information about clock drift

Since it was known from preliminary tests with the ESB nodes that the quality of communication links varies considerably over time, it was decided to repeat periodically the determination of the neighbors of each node and to build a new depth-first tree thereafter. This step was accomplished at the start of every clock hour.

Due to the uncertainty in the radio communication it was decided to use a time triggered scheme in which activities are initiated by the progression of a globally synchronized time-base. Since a considerable clock drift between individual ESB nodes was observed, the sink sends periodically its local time into the network. Table 1 lists the points in time within the span of one hour and the associated actions. All nodes including the sink have the same code; to turn a node into the sink a single flag needs to be set. This considerably facilitated the deployment process. The sink node sent all data over a standard serial port interface to a PC, that stored the data in files.

Time	Action	Result
0	Reset	Resets state of node
1	Start WNX	Nodes send Hello packets, compute link qualities and determine bidirectional links
9	Suspend WNX	List of bidirectional links with link quality
12	Start depth-first search	Depth-first search tree
14	Start measurements	Leaf nodes turn off radio, inner nodes turn off sensors, in intervals of 10 minutes • leaf nodes turn on radio • all nodes send measured data and link states to sink, leaf nodes turn off radio • the sink sends its local time to all nodes in tree

Table 1: Periodic sequence of triggered actions

3.4 The Deployment

The nodes were deployed in a rectangular area with dimensions 140 times 80 meters. The territory was mainly heathland, some spots with taller trees and three smaller buildings. For most pairs of nodes the line-of-sight was obstructed. The majority of the nodes was attached to trees at a height of about 4 meters, some on poles just above the ground surface and 4 nodes were positioned on different floors inside the main building (including the sink). Figure 3 depicts the location of the nodes and of the main building, the sink (square node) and marks the three nodes that failed immediately after the start of the experiment (empty circles). Furthermore, it shows a sample depth-first tree.

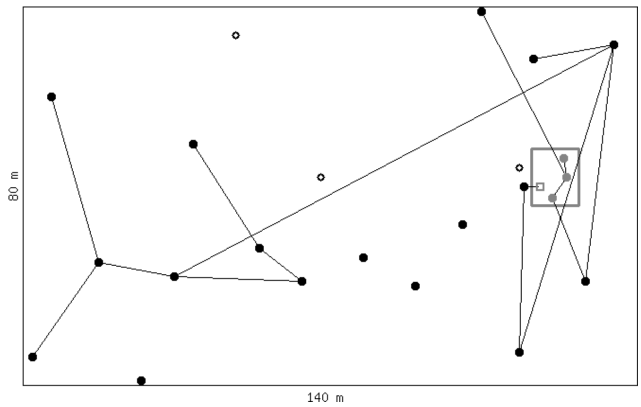

Figure 3: Location of sensor nodes

The software allowed the nodes to operate in two different modes: deployment and application mode. After the nodes were attached to their positions, the software was in deployment mode. Using an additional node attached to a portable computer it was possible to configure the nodes in place by walking through the area. Within the transmission range packets could be sent directly to a node using its identifier. Among other things this allowed

- to query the reachable neighbors of a node (including the quality factors)
- to adjust the transmission range
- to switch into the application mode

The deployment mode made it possible to create a topology according to the experiment's needs. In particular, the numbers of neighbors of a node was limited in order to reduce radio interferences. This approach proved to be useful.

4. ANALYSIS

The following discussion is confined to an analysis of the quality of the communication and the consequences for the application, i. e., the depth-first search. Altogether 24 nodes were used in the experiment, three of the nodes did not send or receive any data after the first day (the reasons are not known). Among the remaining 21 nodes, out of the 210 different pairs of nodes, 45 demonstrably appeared as links in a search tree, but the quality of these links varied considerably.

Link-quality for each neighbor was computed based on the *hello-history*, a bit-field indicating whether or not a hello has been received in the past. Each node kept track of the last 32 expected hellos from each neighbor. Periodically, received hellos – weighted with a coefficient – were summed up to represent link-quality. These coefficients have been selected to be a linear function of the hello-age: more recent hellos were assigned a higher one than very old hellos. To be precise, the latest expected hello was weighted with a value of 11.84, whereas the value of each older hello was reduced by 0.25. Hence, link-quality was an integer between 0 and 255, with the latter being the optimum. In addition, very old hellos had less influence. This approach was chosen to support the suspending phase of WNX: the link-quality must not only indicate how many hellos have been received on the long run, but also how many hellos have been received shortly before.

The application mainly used a unicast transmission scheme with acknowledgments provided by the firmware. In case acknowledgments were not received up to 15 retransmissions were tried. This leads to a maximal latency of 1143 ms, not including the waiting time for channel access. We observed latencies up to 3 seconds. About 50 % of all unicasts were successful. Out of these transmissions the average number of retries was 3.89 with a standard deviation of 2.19. This suggests, that the limit of 15 maximal retransmissions was chosen too high. A lower value may have led to less congestion and may have increased the overall success rate.

Figure 4 depicts the quality values of a link between two nodes inside the building. On the average the quality is above 200 with rather limited variation over time (the following three figures also display the mean value and the standard deviation). Figure 5 depicts the quality values of

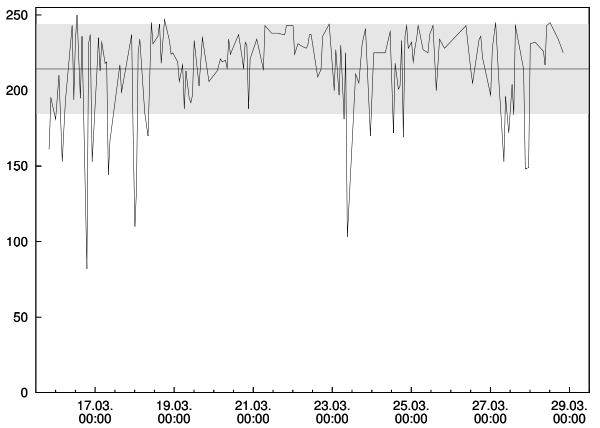

Figure 4: Quality of a link between two indoor nodes

a link between two outdoor nodes. The average link quality is well below 200 with a rather high variation over time. This is a typical example for outdoor links, in some cases the variation was even higher (see Figure 6). Outdoor nodes were on the average much farther apart from each other than indoor nodes. For some links the number of entries in our log file was too low to make a fair judgment. Overall it can be said, that the a priori assessment of the quality of a link is very difficult. We observed cases where a pair of nodes could not communicate despite line-of-sight, while the opposite phenomenon also occurred. Similar to [7] nodes received packets frequently from high quality neighbors, but also occasionally from more remote nodes. Many nodes had asymmetric links, i. e., the link quality in one direction was high, and low in the opposite direction. These links are not used in the application. Moreover, we experienced a correlation between packet size and transmission success: The larger the packets, the lower the probability that the packets reached their destination.

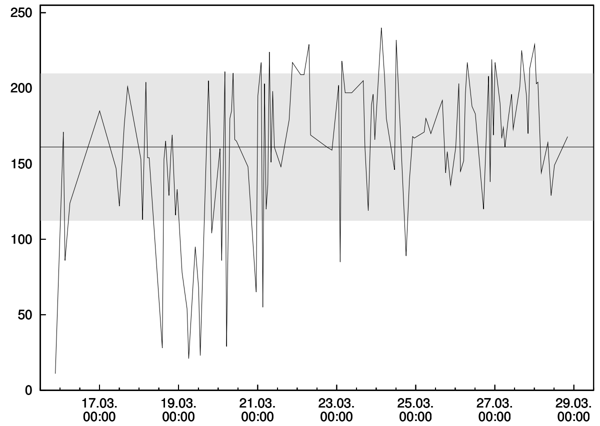

Figure 5: Quality of a link between two outdoor nodes

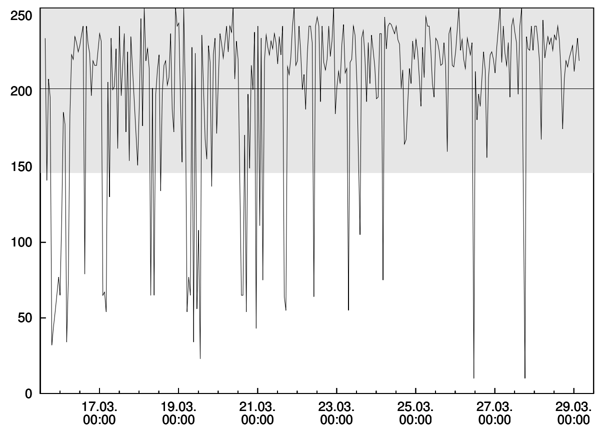

Figure 6: Large variation of link quality over time

Despite the discouraging results about the quality of the links, the depth-first search produced surprisingly good results. The distributed depth-first search was started from the sink 317 times in a period of two weeks. The search successfully terminated in 205 of these cases (in four cases the sink did not reach any node). A successful depth-first search does not imply that all nodes of the network were visited. The largest tree consisted of 19 nodes (over 90 % of all active nodes). Figure 7 displays the distribution of depth-first trees with respect to their size. More than 50 % of all successfully built trees included more than 10 nodes (i. e. 50 % of the nodes). Trees with 17 nodes even occurred 25 times; this is a rather astonishing result, against the background of the quality of links discussed earlier. The trees varied considerably, which is another sign of the changing quality of the links. Not surprisingly, links with high average quality appeared more often as edges of the depth-first tree. The link whose quality is depicted in Figure 4 arose in more than 50 % of all trees.

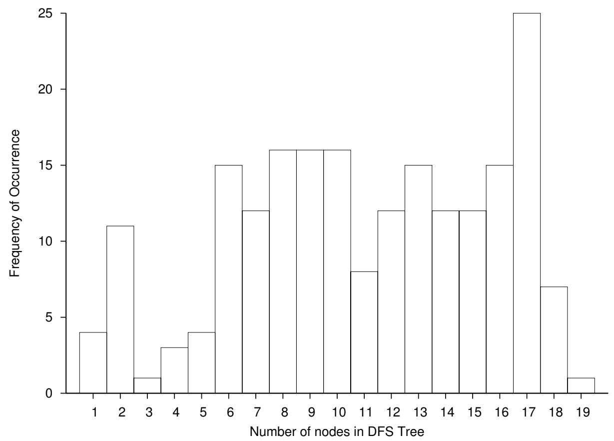

Figure 7: Distribution of total number of nodes in depth-first search trees

The depth-first search trees were used as routing trees, to forward the data measured at nodes towards the sink. Obviously, the successful transmission of a packet towards the root correlates with the depth of the node in the routing tree. Figure 8 displays the relationship between successful delivery of a measurement packet and the depth of the corresponding node in the routing tree. Expectedly, the rate of success drops approximately exponentially with the depth d. The plotted curve $100 \cdot 0.8^d$ closely approximates the delivery rate of the measurement packets. Hence, the average delivery rate can fairly well be predicted based on the average qualities of the individual links. This allows an estimation of the maximal acceptable hop count for multi-hop routing.

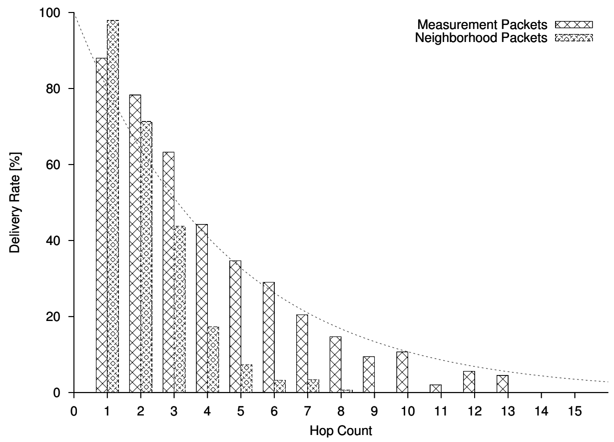

Figure 8: Transmission success in relation to hop count

The transmission success of neighborhood packets exhibits a different run of the curve. The size of a measurement packet is about 70 bytes including the packet header. Packets reporting the neighborhood list to the sink are more than 10 % larger than measurement packets. This leads to a significantly lower success rate as can be seen from Figure 8. The drop of the success rate was much higher than expected: at depth five the success rate is about 7 % compared to 35 % for the smaller packets. The main reason for this is probably not the larger packet size but is related to congestion. Neighborhood packets were sent t seconds after the measurement packets, t randomly chosen between 0 and 15. Due to the retransmission of packets this time difference became very small after a few hops. This explains the lower success rates of neighborhood packets for higher hop counts.

5. CONCLUSION

Like every scientific experiment, real deployments of experimental sensor networks need careful planning and first of all a clear definition of the goals. In sensor networks data logging is almost the only means to acquire data, consequently a logging strategy must be developed in order to collect the data to derive the intended goals. After the deployment there is usually no possibility to intervene in the logging process. The breakdown into a deployment and an application mode proved to be very useful, especially for changing the topology.

As a first conclusion it can be stated, that link quality estimation and neighborhood management are essential to reliable routing in sensor networks. The quality of individual links varies over time for no apparent reasons and unidirectional links of good quality occur more often than bidirectional links of similar quality. This observation suggests, that the concept of unit disk modeling used in many theoretical investigations is not an appropriate model at all. The following lessons can be learned from the experiment: larger packets should be broken up into smaller ones, the number of retransmissions should be modest, the transmissions should be carefully scheduled to avoid congestion, and a good understanding of the implementation is indispensable.

6. REFERENCES

[1] D. M. Doolin and N. Sitar. Wireless sensors for wildfire monitoring. *Proc. SPIE Symp. on Smart Structures & Materials/NDE 2005, San Diego*, Mar. 2005.

[2] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF), RFC 3684, 2004.

[3] ScatterWeb GmbH. The Embedded Sensor Board. `http://www.scatterweb.net`, 2005.

[4] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004.

[5] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *Proc. of the First European Workshop on Sensor Networks (EWSN)*, Jan. 2004.

[6] Y. H. Tsin. Some remarks on distributed depth-first search. *Inf. Process. Lett.*, 82(4):173–178, 2002.

[7] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. First Int. Conf. on Embedded Networked Sensor Systems*, pages 14–27, 2003.