Analysis of a Real Multi-hop Sensor Network Deployment: The Heathland Experiment

Volker Turau, Matthias Witt, and Christoph Weyer
Hamburg University of Technology, Institute of Telematics
Schwarzenbergstraße 95, 21073 Hamburg, Germany

ABSTRACT

This paper reports on the results and 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 the experiment was on the long- and short-term reliability of radio links, estimation of link qualities, the influence of the link quality on multi-hop routing, and on neighborhood exploration.

I. 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 those algorithms has not been implemented on real sensor networks, but evaluated using simulation. Simulations are a valuable and cheap means to evaluate algorithms and protocols, but currently no simulation tool is capable to allow for all imponderabilities of a real deployment of a sensor network in a complex environment over a longer period of time. The significance of such simulations has been questioned recently [1], [2]. The main criticism is on the overly simple radio models in use (such as the unit disc graph model or the model provided by ns-2). In particular the following assumptions present in many simulations must be scrutinized:

- Nodes with the same hardware behave equally,
- a radio's transmission area is circular,
- all nodes have the same transmission range,
- all communication links are symmetric,
- the quality of a link remains constant over time, and
- the area of deployment is free of obstacles.

The first assumption is often not valid for hardware from the low price segment, but networks consisting of several thousands of nodes will probably come from this segment. Small deviations in hardware parts combined with low frequency stability influence the predictability of the radio. The next three assumptions have been disproved by several short-term experiments mostly using regular indoor settings [3], [4], [5], [6]. It is well known that path loss, attenuation, and the presence of obstacles lead to slow fading, i. e., non-stationary error conditions over longer time scales. Link qualities over longer periods of time in realistic settings have not been evaluated systematically. Up to today the number of long-term deployed wireless sensor networks is extremely low compared with the number of publications. There are a few publications reporting about short-term experiments with sensor networks [7], [4], [8]. The most prominent long-term deployment was the experiment on *Great Duck Island* [9], [10]. The VigilNet project reports about the performance of a system composed of 70 MICA2 motes in a realistic outdoor setting [11].

This paper reports about an experiment with a real deployment of a sensor network with 24 nodes conducted in March 2005 in the heathlands of Northern Germany. After the deployment the application ran without any human attention. The goal was to qualitatively assess the deviations from the above assumptions, in particular with respect to a longer period. The results allow an insight into the problems emerging during the deployment and operation of a sensor network and provide valuable information for other installations. They are to a certain degree specific to the particular sensor nodes used, but we believe that many aspects are valid independently of the hardware in use. A preliminary report can be found in [12].

II. THE GOALS

The Heathland Experiment is to our knowledge the first long-term usage of the Embedded Sensor Board (ESB) platform [13]. Foremost, we were interested in the overall performance of the network with respect to the assumptions stated above. In particular the following aspects were analyzed in this experiment:

- General radio performance, symmetry of links
- Short- and long-term stability of links
- Effectiveness of distributed best-link-first algorithm
- Reliability of tree routing

To conduct the analysis, about 6 MB of data was logged per day. Whenever a packet with a sensor reading was sent to the sink, the node's state was included in the packet. Some data was also stored in the EEPROM of each node; this was used for post mortem analysis (e. g., when a node completely failed or was no longer reachable from the sink).

III. THE EXPERIMENT

A. The Hardware

For the experiment, the ESB nodes from the Free University Berlin were used [13]. 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. Each node has 2 KB RAM and 64 KB EEPROM. The nodes were powered by three fresh AA batteries. The sink had a permanent power supply. The power consumption of the nodes according to the specifications of the vendor varies from $8\,\mu$A in sleep mode up to 12 mA when running with all sensors.

B. The Packaging

The nodes had to be prepared for diverse weather conditions including snow, rain, and sunshine. 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, poles etc. using gaffer tape (see Figure 1). Measurements conducted before the experiment proved that the packaging had no noticeable influence on the quality of radio communication. The recorded temperature during the experiment varied considerably with a minimum of $-4\,°$C. Our results show that the applied packaging was appropriate (no node failed due to extraneous cause). The packaging formed an insulation of the nodes and affected the measurements of the sensors. This was accepted, since the sensor data merely served as a payload for the messages.

Figure 1. Node attached to tree

C. 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. Alarming applications follow a different style, where a gateway node gets notified only in case any sensor measures a value above a predefined threshold. The implemented application follows the *sense-and-send* pattern. Nodes periodically sent readings from their five sensors via a routing tree to the sink, which sent all data over a standard serial port interface to a PC, that stored the data. To account for topology changes caused by node failures, decreasing communication radii due to fading battery energy, and moving nodes, neighbor relationships and routing trees were recomputed regularly. This decision was further supported by preliminary tests with the ESB nodes that revealed a big variation of the quality of communication links over time. The application was repeated every 60 minutes at the start of every clock hour. To tolerate the loss of messages a time triggered scheme in which activities are initiated by the progression of a globally synchronized time-base was used. Table I lists the points in time within the span of one hour and the associated actions.

Table I. Periodic sequence of triggered actions

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 BLF	Routing tree
13	Time synchronization	The sink sends its local time to all nodes in 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 again

The overall philosophy of the experiment was to employ self-discovery and self-organization. This means that no infrastructure was provided a priori, the necessary structures for routing were established by the application itself. 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 MAC protocol is based on a non-persistent CSMA protocol, not considering the hidden terminal problem. The link layer provides unicast transmission based on Automatic Repeat Request using positive acknowledgments. The protocol discards a packet after 15 unsuccessful trials of submission. The link management includes a neighborhood discovery protocol called *Wireless Neighborhood Exploration* (WNX). It is a modified and extended implementation of TND, the proactive neighborhood discovery protocol of TBRPF [14]. WNX determines uni- and bidirectional links and adds a quality descriptor to every link. Using the bidirectional links provided by WNX, a distributed algorithm called *best-link-first* (BLF) builds a routing tree rooted at the sink, all unidirectional links are discarded. BLF is a distributed depth-first search where high-quality links are preferred over low-quality ones, the implementation is based on [15]. Each node maintains an ordered list with the identifiers of the successors and the identifier of the parent node. The tree is used to route messages from nodes to the sink and vice versa. 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 and BLF, 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, while inner nodes do not turn off their radio at all.

Upon the completion of a BLF run all nodes send their neighbor lists including the quality values 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 sent periodically to the sink:

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

Periodic repetition of the application and permanent power supply of the sink were chosen to provide the fault tolerance needed to run the experiment unattendedly. Since a considerable clock drift between individual ESB nodes was observed, the sink sends periodically its local time into the network.

D. The Deployment

The nodes were deployed in a rectangular area with dimensions 140 times 80 meters. The rugged 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). The complete setup occupied 5 persons for a full day. Figure 2 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 routing tree. Note that the route leaves the building only to return to a node inside the building after four hops and finally leaving it again. This already gives a flavor of the relationship between link quality and distance.

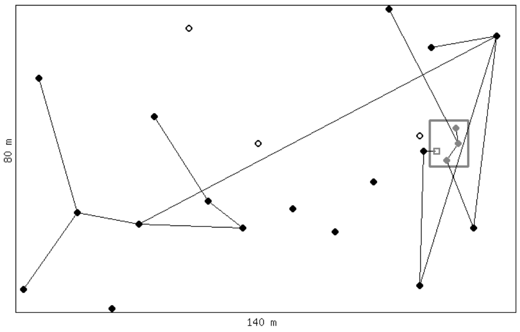


Figure 2. Location of sensor nodes

E. Neighborhood Protocol

A prerequisite for most routing algorithms is that every node must be aware of its direct neighbors. Since the grades of wireless communication links considerably depend on many factors which are hard to predict, the neighbors of each node have to be determined online. The reception of a single message is not an acceptable indicator for a useful link. It is necessary to observe packet success and loss over a longer period of time and to estimate the quality of the link, but resource restrictions do not allow to keep an extensive link history. Instead a probabilistic perspective is applied. Larger message histories allow a long-term characterization of the quality of a link but also require more storage for every link of a node. Due to the lack of efficient MAC layer protocols that work with unidirectional links (e. g., both RTS-CTS and ACK based schemes cannot be used directly), a link estimate must consider both communication directions of a link. As a consequence link estimates must be exchanged. The challenge is to find a distributed algorithm for link estimation that operates within the tight resource boundaries of sensor nodes. Once useful links are determined, higher-level protocols can build a routing infrastructure on the resulting graph.

The implemented protocol WNX is an extension of TND as defined in RFC 3684 [14]. The assessment of links is based on the receive rate of periodically broadcasted HELLO messages. This approach was selected, because the alternative of passively snooping application packets on the channel requires continuous traffic. Since there was no application traffic at all when WNX was running, this was no valid alternative. WNX is exclusively based on broadcasting messages. To reduce the probability of collisions of HELLO messages of different nodes, a random jitter was introduced. Each HELLO message contains a sequence number, which is incremented for every message. These numbers are used to detect lost messages. A history of h HELLO messages per link is maintained using h bits of storage. Based on this data, links are classified as LOST, 1-WAY, or 2-WAY. Upon the arrival of the first message over a link, it is rated as LOST. It is promoted to 1-WAY if k out of the last n messages are received ($n \leq h$). A 1-WAY link becomes a 2-WAY link if it has the status 1-WAY at both ends of the link. In case one of these prerequisites is no longer satisfied, the link is downgraded to status LOST. When the status of a link changes, this information is included in the next HELLO message, as a consequence this packet has a larger size. Agility and stability of WNX is controlled by:

1) Span of time T_H between consecutive HELLO messages
2) Thresholds for transitions of link states

Reducing T_H leads to a more agile behavior needed for example in a mobile environment. The downside is increased traffic which comes along with higher energy consumption and a higher collision risk. The second criteria has to do with stability. Choosing low threshold values leads to frequent changes of link states if messages are periodically lost. This instability complicates routing considerably, the routing topology is unlikely to stabilize and routing problems, such as cycles and stranded nodes, will be common. A high threshold can lead to a communication infrastructure with too few links. WNX was used in the Heathland Experiment with $h = 32$, $T_H = 5$ seconds and a maximal jitter of 0.1 seconds. A link is promoted to a 1-WAY link if at least 8 out of the last 16 HELLO messages were received. Selecting optimal parameters for WNX based on environmental factors such as network density is subject of future research.

Each node maintains a list with estimates of link qualities. Bounded memory allowed only lists of length eight (requiring 144 Bytes), this entails the danger that a link is not included in the list because it is already full. This can lead to the situation that in deployments with high node densities a physically connected network is disconnected from the routing perspective (see Section IV-D). A problem is that the message history is not recorded if a node is not in the neighbor list. The implemented list eviction policy aimed to increase stability, only links with status LOST are discarded. If there is no link with status LOST, no new links are inserted. A link becomes LOST, when the node does not receive any out of the last six HELLO messages, and a link with status LOST is finally purged if there has no HELLO message been received within the last 30 seconds. Thus, the eviction of a failed 1-WAY resp. 2-WAY link is enforced after maximal 60 seconds. Reducing this value in a deployment with high message loss bears the danger that link changes occur very often, making routing extremely difficult. On the other hand, if a new node is activated it takes other nodes at least 40 seconds to discover a 2-WAY link to the new node.

The WNX protocol ran at the beginning of every hour for 8 minutes and then link qualities were frozen for the rest of this hour. To terminate WNX it enters a suspend mode. This was done to enable a node to distinguish the case that another node stopped sending HELLO messages from the case that all HELLO messages from a node are lost. Without this mechanism, link qualities could be tampered due to differences in termination time caused by clock variances.

F. Link Quality

WNX determines for each node a limited number of bidirectional links. To get a reliable structure, the *best* links should be preferred. To meet this end, a numerical quality is calculated for each unidirectional link. Estimating link qualities is independent from neighbor discovery. The goal is to find an estimator that accurately represents link qualities, reacts quickly to changes in link quality, yet is stable, has a small memory footprint, and is simple to compute. Due to results of Zhao and others [8], [6] we decided not to use signal strength as the basis to evaluate link quality. Instead the computation is based on the message history, the procedure is a variant of *Time Weighted Moving Average* as proposed in [16]. Link quality is the weighted sum of the received messages, the weights have been selected to be a linear function of the age of the message: more recent hellos are assigned a higher weight than very old hellos. In more detail, let $b_{k,i} = 1$ if the i-th message of the actual time window was received over link k, and $b_{k,i} = 0$ otherwise. The weights of two consecutive messages differ by Δ. This leads to the following formula to calculate the quality Q_k of link k for this time window:

$$Q_k = \sum_{i=0}^{h-1} b_{k,i}(C - \Delta i)$$

The values of C and Δ were chosen such that the link quality is an integer in $[0, 255]$, with 255 being the optimum. This is satisfied if $C = (255 + h'\Delta)/h$ and $\Delta \in [0, 255/h']$ with $h' = h(h-1)/2$. To simplify the calculation of the qualities, $\Delta = 0.5$ was chosen. The computation can be implemented very efficiently. Apart from some local variables, WNX requires only 158 Bytes of storage in each node in case $h = 32$.

IV. ANALYSIS

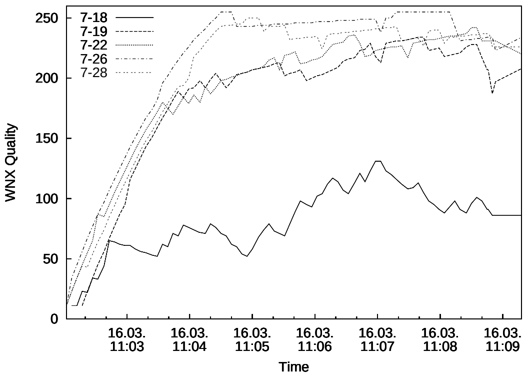
A. Neighborhood Protocol and Link Quality

To analyze our neighborhood management policy the state of the table of each node at the end of every WNX phase was sent to the sink. The data logged by the sink shows that in more than two thirds of all instances the tables contained between 4 and 6 links. Only in 6.6 % of all cases completely full tables occurred. This situation was further investigated. On the average 2.5 links in the full tables had status LOST, but all tables had at least two 2-WAY links. It is therefore reasonable to assume that reducing the size of the table to 7 or even 6 entries would not change the connectivity of the network. It remains an open problem to find an ideal table size for a given node density. Another eviction policy might take topology information into consideration in order to prevent the described phenomenon.

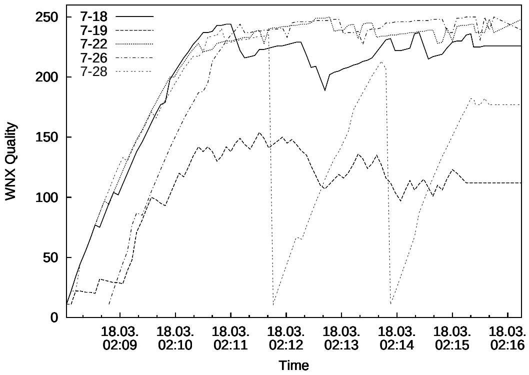
Figure 3 depicts the course of the link qualities after the start of WNX for node 7 at two different points of time (16th and 18th of March). In every case there are 5 links in the neighborhood table. After about 2 minutes (i. e., about 24 HELLO messages) the link qualities reach a stable level. In Figure 3(a) all but one link reach link qualities above 200. The link from node 7 to node 19 has very good quality during the first period but low quality during the second period. For the link 7 to 18 this behavior is inverted.

Figure 3 suggests that link qualities are relatively stable over time, but this was not the case. Figure 4 depicts the qualities of three different links over the period of two weeks. The figures also show the average quality and the standard deviation. The qualities displayed in Figure 4(a) belong to a link connecting two nodes inside the building. On the average the quality is above 200 with rather limited variation over time. In general links between outdoor nodes had a lower average and a higher variation over time. Figure 4(b) and (c) show two typical cases, in some cases the variation was even higher. Our results show that the success rate of packet delivery only dropped impalpably during one hour. In the light of the results depicted in Figure 4 the optimal period of recalculating the neighborhood relation remains an open problem.

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 based on distance, line-of-sight, and environment 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 [4] nodes



(a) Low quality link to node 18



(b) High quality link to node 18. Node 28 crashed twice due to firmware problems and restarted itself.

Figure 3. Link qualities during the first 8 minutes of a WNX phase as seen by node 7

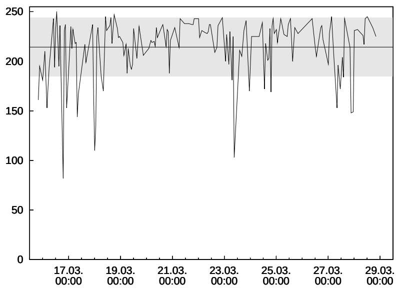
received packets frequently from high quality neighbors, but also occasionally from more remote nodes.

B. Asymmetric links

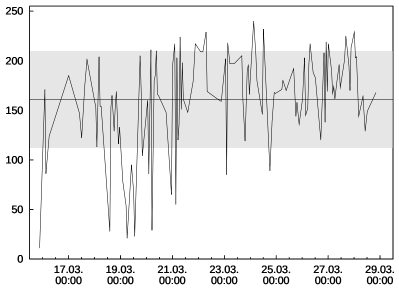
Many nodes had asymmetric links, i. e., the link quality in one direction was high, and low in the opposite direction. Figure 5 displays two courses of link qualities seen from opposite ends of a link. In Figure 5(a) both directions have roughly the same quality over the complete time, whereas in Figure 5(b), one direction has a much higher quality as the opposite direction. Our experiments support the commonly held belief that link asymmetries are due to differences in hardware calibration.

C. Retransmission Policy

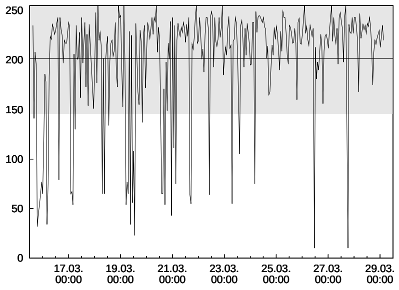
The firmware of the ESB nodes does not include virtual carrier sense mechanisms like CTS and RTS for hidden-terminal mitigation. To improve link reliability, per-hop retransmissions at the MAC layer are employed in the unicast transmission scheme. In case acknowledgments are not received up to 14 retransmissions are tried. The unicast scheme was used across all phases of the application except the initial WNX phase. The downside of this policy is increased traffic and increased power consumption.



(a) Two indoor nodes



(b) Two outdoor nodes

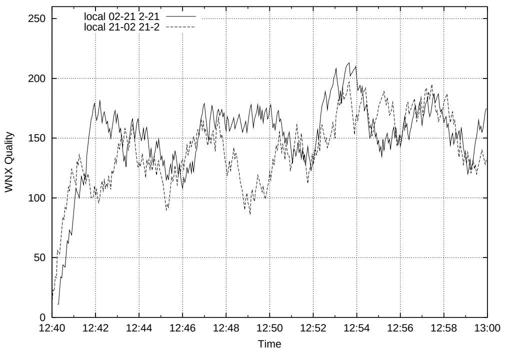


(c) Two outdoor nodes

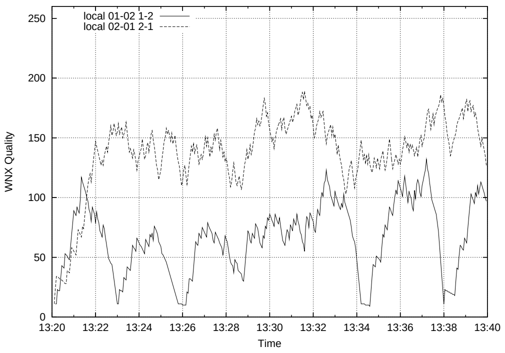
Figure 4. Course of link qualities over two weeks

About 50 % of all unicasts were successful, thus for each of the 50 % unicasts that were unsuccessful 15 transmissions were tried in vain. This seemed to be an enormous waste of energy. To further analyze this situation the successful transmissions were considered in detail. Figure 6 displays a histogram of the retransmissions for these cases. The average number of retransmissions was 3.89 with a standard deviation of 2.19. This suggests that the limit of 14 maximal retransmissions was chosen too high, other protocols use lower values (e. g., SMAC implementations often use 5–7 retransmissions [17]). Reducing the number of retransmissions to 8 or 7 would have considerably reduced the number of messages and consequently the likelihood of collisions would have declined. This would have ultimately increased the overall success rate and helped to save energy.

It would have been desirable to be able to distinguish between packet losses due to collisions of simultaneous transmissions from those caused by environmental factors. But the recorded data did not allow such an analysis.



(a) Symmetric link



(b) Asymmetric link

Figure 5. Link quality as seen from opposite directions

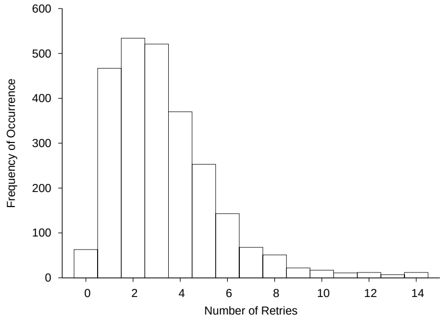


Figure 6. Retries of successfully delivered messages

D. Best-Link-First Algorithm

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 routing tree. 15 of these links constitute the core of the deployment because they contribute two thirds of all links in all trees. Despite the discouraging results about the quality of the links, BLF produced surprisingly good results. The algorithm was started from the sink 317 times in a period of two weeks and successfully terminated in 209 of these cases. A successful BLF run does not imply that all nodes of the network were visited. It merely means that all nodes reachable from the sink were visited. Hence, the network was not connected with respect to the links determined by WNX. There are two kinds of causes for this phenomenon:

- Due to varying link reliabilities the network was temporary disconnected.
- Sometimes a node reset itself shortly before the end of a WNX phase, then the remaining time was too short to establish new 2-WAY links.

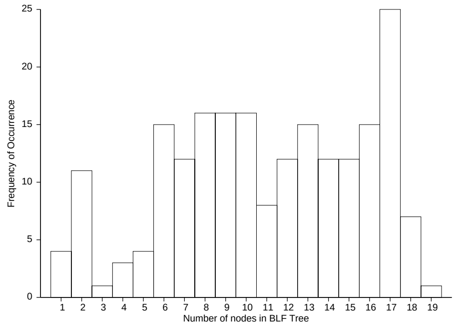


Figure 7. Distribution of BLF tree size

As stated earlier the limitation of 8 neighbors per node was not the reason for the disconnectness. The largest tree consisted of 19 nodes (over 90 % of all active nodes). Figure 7 displays the distribution of routing trees with respect to their size. More than 50 % of all successfully built trees included more than 10 nodes (i. e., 50 % of the active 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 for the changing quality of the links over time. Not surprisingly, links with high average quality appeared more often as edges of the routing tree. The link whose quality is depicted in Figure 4(a) arose in more than 50 % of all trees. Figure 7 also shows that the network of the 21 active nodes was never fully connected. Notwithstanding the big variance in routing trees, there was also a constant element. The list of each node's successors was relatively stable over time. If S_i is the set of all successors node i had during the experiment, then the average size of S_i over all nodes is 3.8 and the median 4. Thus, despite changing link qualities there is a stable element which can be exploited at higher protocol layers. For example, the 25 most often used links constitute more than 90 % of all links in all routing trees. These cognitions may lead to a reliable service on top of the non-deterministic link behavior.

The usage of maximal 14 retransmissions leads to a maximal latency of 1143 ms for sending a packet from one node to another, not including the waiting time for channel access. Latencies up to 3 seconds were observed. Referring to last section's results, a successfully transmitted message needed on the average about 300 ms. Figure 8 depicts the average durations of all BLF runs with a fixed number of nodes. The number of messages sent by the algorithm is between $2m$ and $4m - (n - 1)$, where m denotes the number of 2-WAY links and n is the number of nodes.

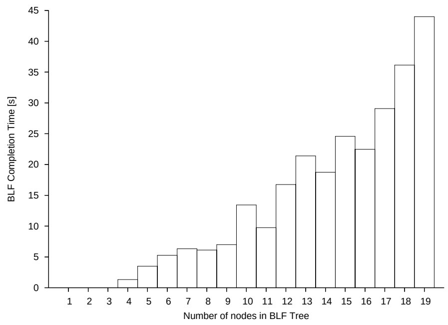


Figure 8. Average durations of BLF runs

E. Reliability of Data Delivery

The BLF 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 9 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.

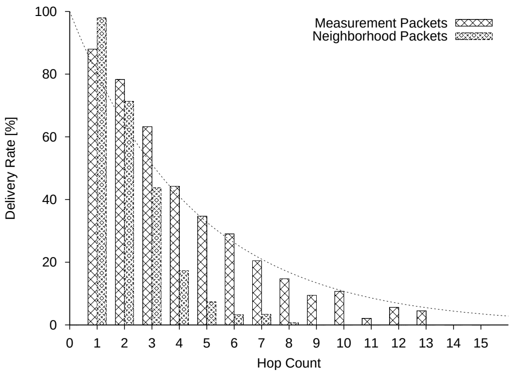


Figure 9. Transmission success in relation to hop count

The transmission success of neighborhood packets exhibits a different run of the curve, the success rate is considerably lower. The reason for this drop is rooted in the scheduling of these packets and the resulting congestions. 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. Thus, these packets obstructed the queues and were eventually discarded (the size of the queue is about 400 Bytes only). This explains the lower success rates of neighborhood packets for higher hop counts.

This behavior raises the question of scalability of this tree style of data collection in wireless networks. If the number of successors of each node is increased, the likelihood of collisions due to messages sent from nodes to their fathers also increases. If the depth of the tree is increased, messages queue up in nodes and are discarded if queues overrun. Due to the distributed nature of the setting, intelligent timing schemes are difficult to realize. Hence, alternative strategies such as in-network aggregation must be deployed.

V. DISCUSSION

Because the ESB platform has not been used in a long-term outdoor experiment, there was no documented experience with respect to this kind of usage. A serious problem faced during the experiment was that the nodes only slowly adopt to changes in temperature. The nodes were prepared for deployment at room temperature, whereas the temperature outside was about 5 °C on that day. This meant that the radio communication of nodes did not work at all directly after deployment for some time, this caused a lot of confusion. The packaging of the nodes did prevent sharp changes in temperature, so this had no influence during the experiment.

While testing the software for the application, we noticed that the firmware experienced occasional internal resets. Due to a lack of a proper debugging environment and rather sparse documentation, it was difficult to track the causes of these resets. During this process a few bugs in the firmware were discovered and fixed. This experience led to the adoption of a time-triggered architecture as opposed to the initially envisaged communication-style architecture, so at least at the start of every hour the same conditions held. Bugs in the firmware or misconceptions about its functionality can lead to serious misinterpretations of results.

After 16 days the batteries of the nodes near the sink ran out of energy. After the experiment we learned that the access to the real time clock was rather energy intensive, the clock is accessed via a 2-wire serial data bus requiring the setting of registers for accessing the bus, followed by writing control and start bytes and then reading 7 Bytes from the bus. Since every packet contained a time stamp, these accesses occurred frequently and were probably one of the main reasons for the short life time.

The application software was transferred from a laptop to the nodes using a cable, this procedure took about 1 minute per node. This approach is no longer feasible if the number of nodes is one magnitude higher, so other technologies with better scalability are needed. The provided tool to flash the code over the air could be a solution, but at the time of the experiment this did not work reliably.

In retrospective the decision to send all data to be used in the analysis of the experiment through the network to the sink needs to be reevaluated. An advantage was that the data was instantly available for monitoring the experiment. The disadvantage was that due to the error rate of about 50 % a lot of this data never reached the sink and consequently was not available for the concluding analysis. A minor part of the data was written into the EEPROM of each node, this data did not suffer this problem. But the size of the EEPROM did not allow the storage of larger amounts of data. For future experiments it is planned to write more data to the EEPROM and periodically collect this data to allow the reuse of this storage. The disadvantage is that the application needs more attention and reading the data via radio consumes energy.

Using a time triggered architecture in conjunction with radio communication caused some problems as explained above. The delays resulting from retransmissions or medium access time are hard to predict. The precaution used in the experiment was to use a very generous timing scheme. Using a fixed scheme does not give consideration to the scalability of an application. And unless the reliability of communication links increases, alternative solutions must be searched for. The applied procedure to synchronize clocks also suffered from the low success rate of unicast messages, this also calls for a different solution.

VI. CONCLUSION

Real deployments of sensor networks are very expensive, a careful planning and a clear definition of the goals is absolutely necessary. After the deployment there is usually no possibility to intervene in the experiment. The Heathland Experiment analyzed wireless communication in an outdoor environment over a longer period of time. Firstly, it strongly confirmed earlier findings that simple radio models used in many simulations, such as the unit disc graph model, do not properly represent the reality. The experiment demonstrated that the quality of individual links remains relatively stable over short periods of time but varies considerably in the long run and that unidirectional links of good quality occur more often than bidirectional links of similar quality. We believe that the obtained results are also valid for other hardware from the same price segment. The combined effects of low frequency stability, variable environmental conditions in the field, and slow fading due to presence of obstacles cannot be reproduced in currently available simulation tools. The required adaptive protocols can only be evaluated in field trials.

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. The real challenge of energy efficient wireless sensor networks is to create a reliable behavior on top of a non-deterministic communication layer. The results of the Heathland Experiment are an indication that this requires cross-layer optimization, i. e., a joint optimization of the MAC, link, and network layer.

REFERENCES

[1] D. Kotz, C. Newport, and C. Elliott, "The mistaken axioms of wireless-network research," Dartmouth College, Hanover, NH, Tech. Rep. TR2003-467, July 2003.

[2] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: the incredibles," *SIGMOBILE Mob. Comp. Comm. Rev.*, vol. 9, no. 4, pp. 50–61, Oct. 2005.

[3] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," UCLA, Tech. Rep. CSD-TR 02-0013, 2002.

[4] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. 1st Int. Conf. Emb. Networked Sensor Systems*. New York: ACM Press, 2003, pp. 14–27.

[5] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *The 1st IEEE Int. Workshop on Sensor Network Protocols and Applications*, Anchorage, USA, May 2003.

[6] D. De Couto, D. Aguayo, B. Chambers, and R. Morris, "Performance of multihop wireless networks: shortest path is not enough," *1st Workshop on Hot Topics in Networks*, Oct. 2002.

[7] O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan, "Interaction of Retransmission, Blacklisting, and Routing Metrics for Reliability in Sensor Network Routing," in *Proc. 1st IEEE Conf. on Sensor and Ad Hoc Comm. and Networks*, Santa Clara, CA, Oct. 2004.

[8] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st Int. Conf. on Embedded Networked Sensor Systems*. New York: ACM Press, 2003, pp. 1–13.

[9] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," in *Proc. 1st Eur. Workshop on Sensor Networks*, Jan. 2004.

[10] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," *Proc. 2nd ACM Conf. on Emb. Networked Sensor Systems*, Nov. 2004.

[11] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh, "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM Trans. on Sensor Networks*, 2006.

[12] V. Turau, C. Renner, M. Venzke, S. Waschik, C. Weyer, and M. Witt, "The Heathland Experiment: Results And Experiences," in *Workshop on Real-World Wireless Sensor Networks. SICS T2005:09*, Stockholm, June 2005.

[13] ScatterWeb, `http://www.scatterweb.net`, 2006.

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

[15] Y. H. Tsin, "Some remarks on distributed depth-first search," *Inf. Process. Lett.*, vol. 82, no. 4, pp. 173–178, 2002.

[16] A. Woo, "A Holistic Approach to Multihop Routing in Sensor Networks," Ph.D. dissertation, University of California, Berkeley, 2004.

[17] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493–506, 2004.