Field Trials with Wireless Sensor Networks: Issues and Remedies

Volker Turau Matthias Witt Marcus Venzke
Hamburg University of Technology, Institute of Telematics
Schwarzenbergstraße 95, 21073 Hamburg, Germany

Abstract

Extensive field trials are essential to evaluate protocols and algorithms for wireless sensor networks. The high costs of such trials demand for a systematic approach. This paper discusses the issues of setting up a field trial such as packaging, logging, and deployment. Details about a concrete field trial to evaluate a topology discovery algorithm are presented.

1. Introduction

Wireless sensor networks (WSN) are networks of small, resource-constrained wireless devices equipped with sensors embedded in a dynamic physical environment. The field gained momentum around the year 2000 with the availability of relatively inexpensive nodes, sensors, and radios [1]. Projects like *Smart Dust* provided the motivating vision for this research. In principle WSNs are distributed systems, but they pose some unique challenges, foremost resource limitations, high failure rates, and ad hoc deployment. Within the last years many algorithms and protocols have been proposed to solve the problems of WSN. The vast majority of these 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 WSN in a harsh environment over a longer period of time. To attain a deeper insight into WSNs, experiments with real deployments are indispensable. But up to today the number of long-term deployed WSNs is extremely low compared with the number of theoretical results. Extensive field trials require a lot of resources and are very time-consuming. Moreover, systematic approaches for WSN deployment are missing today. While algorithms, protocols, and architectures have been investigated, not much has been achieved in the area of deployment support, pre-deployment debugging, or even a methodology that allows for reliable monitoring of WSNs.

This paper discusses the issues of setting up and running a field trial for WSN in general. In the second part details about a concrete field trial to evaluate a topology discovery algorithm for WSNs are presented. A detailed analysis about the results of this trial can be found in [2].

2. Related Work

A driving application area of WSN research is environmental monitoring. The vision is to revolutionize biological and climate monitoring, by providing instant data at granularities unrealizable by other means. The majority of field trials of WSNs have this application background. One of the first larger field trials was the habitat-monitoring system on Great Duck Island [3], [4]. The WSN collected data about underground nesting burrows and surface micro-climates such as temperature, humidity, and occupancy. The data was used to correlate nesting patterns with micro-climates. Two different networks were deployed, a 50-node single-hop network and a 100-node multi-hop network covering an area of 200×100 m. Nodes periodically sampled data and routed it to a device equipped with a high-gain antenna to transfer the data to a distant solar-powered laptop with satellite Internet link. Apart from a few parameters that could be tuned in situ, the nodes were completely preconfigured. To verify the collected data, an independent verification network was installed utilizing conventional technologies. This network collected short movies using in-burrow cameras for the final verification process.

The sensor networks deployed in field trials are often restricted in some way: single-hop network, artificial environments, very small networks. Römer et al. give a review of several deployments [5]. Apart from primarily application-oriented trials there were also field trials focusing on link stability and network protocols. Woo et al. conducted a trial with a 50-node network placed inside a building [6]. The short experiments were focused on link estimation, neighborhood table management, and reliable routing protocol techniques.

EmStar [7] is a software framework for WSN deployment support. It consists of several tools for simulation, emulation, and visualization of applications. Also hybrid modes for combination of simulation and real wireless communication are available. A set of libraries and services exists for plugging them together in custom applications. EmStar is aimed at the high-end spectrum of WSN platforms, namely 32-bit embedded microservers.

3. Setting up a Field Trial

Outdoor field trials of WSNs pose problems that do not occur in lab experiments. Some of the reasons are harsh environmental conditions (e. g. low temperatures), long-term unattended operation, complex deployment process, connection to the wired world, and large number of nodes. This section discusses the set up of a field trial of a WSN and evaluates different alternatives.

3.1. Hardware

There is a wide range of sensor node hardware available, ranging from small devices with 8-bit microcontrollers as CPUs, 2–100 KB of working memory, 64–1024 KB of flash secondary storage, and a low-power radio operating at 10 kbit/s up to larger devices with 32-bit CPUs, megabytes of working memory and secondary storage, and supporting the Bluetooth wireless standard, such as the Intel Mote [8]. Obviously, the performance of an algorithm will strongly depend on the selected hardware, especially the quality of the transceiver and communication technology is of high importance. Thus, the generality of the results of a field trial is restricted by the hardware. Because our research aims at large-scale networks, we concentrate on the low end of the hardware spectrum. This implies that applications have to deal with very unreliable communication links and have to be content with minimal resources. For this reason, the results of the field trial are valid for this class of hardware, generalizations to a wider range of hardware can only be drawn after a careful analysis.

3.2. Packaging

Outdoor applications present an additional set of challenges not seen in indoor experiments. Most sensor node hardware is still of prototypically nature and does not come with a packaging suitable for outdoor deployment. To withstand variable weather conditions, protective packaging that minimally obstructs the sensing and communication functionality has to be provided. In most cases custom enclosures are too costly. If the accuracy of sensor values is not of importance, off-the-shelf rain-proof boxes with a rubber seal, antennae inside, wrapped in plastic are sufficient. The effect of the packaging on the quality of the communication must be analyzed prior to deployment.

Another aspect that is usually not present in indoor experiments is the variation in temperature during the trial. Temperature directly affects battery voltage, which in turn affects radio communication. Also, some components may not operate as expected at low temperatures (e. g. hardware clocks, digitally controlled oscillators, sensors).

3.3. Parameters of Interest

The parameters of interest to be observed and logged in the field trial have to be determined in the experiment's planning phase. Exemplary parameters are number of sent and received packets, number of retransmissions, energy consumption, or sensor values. The parameters are the basis for evaluating the field trial and to recognize and analyze problems during the execution for improving subsequent trials. Logging as much data as possible seems to be desirable, especially because repeating the field trial is time-consuming and costly. Unfortunately the quantity of data that can be logged is restricted by memory limitations of the nodes, energy consumption for radio transmissions, and the available network bandwidth. Thus, extent and importance of parameters have to be balanced with the effort for logging. The importance is derived from the goals of the field study.

The amount of logged data can be significantly reduced if parameters are aggregated inside the network by the nodes. Aggregation includes statistic evaluations (e. g., average, maximum, number of exceedings of a threshold) and the logging of data only if a specific situation is recognized (e. g., an event of interest occurred, faults). This shifts the interpretation partially into the WSN. This process runs the risk of losing original data that may be needed for the analysis of unpredictable events. Logging raw data provides more potential for subsequent evaluations (e. g., feeding the data into a simulation tool) and for analysis of problems that occurred during the field study.

Further considerations cover the logging of meta-data for the parameters of interest. When logging events, knowledge about their logical order might be sufficient. Some analyses might require time stamps from a real-time clock, but their significance is limited due to the lack of clock synchronization between different nodes.

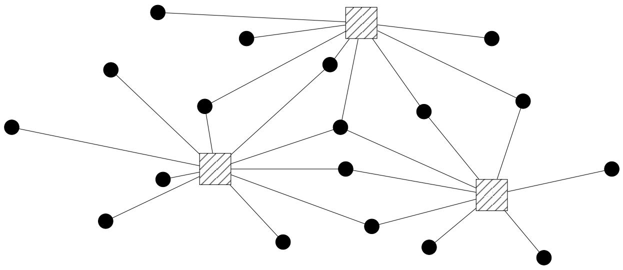
3.4. Logging Strategy

The data measured while running the trial has to be made available to evaluation software executed outside the WSN. There are six aspects to be considered:

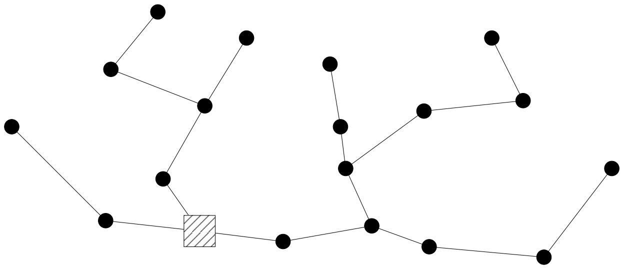
- Completeness: Is the measured data available without loss?
- Availability: When is the data available for evaluation?
- Scalability: Is the strategy still applicable when the number of nodes grows?
- Time and effort: How complex is the deployment of the logging hardware and the collection of the data?
- Energy consumption: How much energy is spent for data retrieval?
- Quantity: How much data can be stored?

There are two alternatives for the storage: locally in the non-volatile memory of each node or outside the network in a persistent storage. The first option has the advantage that the data is not lost in case a node becomes disconnected from the rest of the network, but bounded storage resources limit the quantity of data to be logged considerably. Logging units such as power-supplied devices with large memory can be a way out at the cost of higher energy consumption. The data is regularly transferred to these units. The transfers must be sufficiently reliable to avoid data loss. Figure 1 (a) shows an example with three logging units; every node broadcasts its data, whereon all logging units that receive it store the data in persistent storage. Duplicates can be discarded while merging the data before evaluation. To compensate for data loss, a retransmission scheme based on acknowledgments can be introduced. The precondition for this approach is that each node has a logging unit within its transmission range. A big disadvantage is that the data is only available after the end of the field trial, thus a continuous monitoring of the operation of the system is not possible. A remedy can be the periodic collection of the data by manually connecting a portable storage device to the nodes. This can be impracticable due to the packaging of the nodes or their reachability.

To store the data outside the network, it must be transported to the storage device through one or more gateways. These gateway nodes are either physically connected to the *outside world* or use another communication channel such as GSM (Figure 1 (b) shows an example with one gateway). The setting of the trial usually restricts the number of gateways in the first case. In the second case more complex hardware is needed and the communication requires considerably more energy. The data of nodes not connected to a gateway needs to be delivered to a gateway, thereby data can be lost due to the unreliable nature of the transport. This data is not available for analysis. Because many trials have reported success rates as low as 50 %, this can seriously decrease the value of the trial.



(a) Wireless logging without routing; the squares represent the logging units



(b) The same topology using routing on a spanning tree

Figure 1. Different logging strategies

Since large-scale trials are relatively costly, it may be worth taking measures to reduce data loss. Nodes forwarding data to a gateway node can buffer data into local non-volatile memory until the storage is confirmed by a gateway. This buffering can mitigate times of disconnectness at the added expense of buffering packets along the path. Also accumulated data can be compressed leading to an energy reduction for the transport. The buffering in flash memory comes with an additional energy consumption.

Logging data in WSNs can be automated or performed manually. Automatic logging has the advantage that it runs without attention, so no intervention is needed during the experiment. Manual logging does not require stationary logging units, to guarantee completeness of data a high degree of discipline is needed. While the post mortem analysis usually provides the best insight into the workings of the protocols and algorithms, the value of online monitoring cannot be overestimated. Discovering errors or failures at the earliest point in time can save a lot of time and enables the intervention in the trial to correct software or replace hardware. Another option is a hybrid approach whereby only the most essential data for a post mortem analysis is stored locally, while additional data is logged through a gateway. Table 1 summarizes the pros and cons of the logging strategies.

Table 1
Classification of logging strategies (C: Completeness, A: Availability, S: Scalability, T: Time and effort, E: Energy consumption, Q: Quantity)

Logging strategy	**C**	**A**	**S**	**T**	**E**	**Q**
Data is stored in non-volatile storage of nodes	+	−	+	−	0	−
Every node is physically connected to a logging unit	+	0	−	−	+	+
Data is sent to a logging unit in transmission range	+	0	−	0	0	+
Data is routed to nearest gateway	−	+	+	+	−	0

3.5. Long-term Consideration

Field trials of WSNs are applications that are supposed to run unattendedly for weeks or months. Hence, the robustness of the network is of high importance. The overall robustness of a sensor network relates to hard- and software. It depends on several independent factors: ability of node software to recover from errors, how well nodes withstand harsh outdoor conditions, lifetime of battery power sources, etc. In case the system does not perform reliably in any of these areas, it may fail to deliver the data needed for evaluation.

To protect against permanent node failures in multi-hop environments (e. g., due to energy dissipation, physical damage, environmental interference), the routing procedure must not rely on static structures. This requires dynamic routing algorithms, these may cause additional communication with negative effects on energy consumption. Furthermore, storage requirements of complex algorithms may exceed the available resources. An alternative consists of a periodically repeated application. Node crashes or failed communication links will interrupt the overall execution only until the beginning of the next period. This suggests a time triggered scheme in which activities are initiated by the progression of a globally synchronized time-base. Such a scheme requires the radios to maintain synchronized clocks. To allow a correct interpretation and correlation of sensor reading, this is usually required anyway. It is not necessary for all nodes to have the same global clock, but local variations among the links should be small enough to ensure synchronization at a global level. There are several proposals for time synchronization protocols in sensor networks [9]. The needed accuracy has to be balanced against the generated overhead in form of more network traffic and additional energy consumption.

3.6. Deployment

The final step before the start of the experiment consists of the deployment of the nodes in the field. In some cases legal issues have to be considered with respect to the usage of wireless communication. Deployment is a one-time activity, changes after the deployment are often very costly if not impossible. Before the actual deployment, lab trials should be performed to calibrate characteristics of radio transmission, configuration parameters of protocols etc. Software tools like EmStar [7] can be a great help. Approaches to deploy WSNs in lab setups use cables for program download, control, and monitoring. This approach is limited due to scalability issues and usually infeasible for deployment in the field. Downloading the code to all nodes in parallel using wireless communication provides the needed scalability. Transmission errors and per node settings (e. g., the assignment of an ID) often make a manual handling necessary. To support the deployment process, field tools that run on small, PDA-class devices can be very helpful. By carrying them through the field, they can be used to run a final self-diagnostic program to verify the node's health, to explore and adjust network neighborhood (e. g. by altering the radio transmission power), and for placement control.

4. The Heathland Experiment

This section reports on the design and experience gained during a field trial of a WSN in the heathlands of Northern Germany in March 2005. The goal of the experiment was to evaluate a new topology discovery algorithm with a focus on estimation of link qualities, the influence of the link quality on multi-hop routing, and on neighborhood exploration. The trial was scheduled to run until the batteries were empty.

4.1. The Topology Discovery Algorithm

The high number of nodes, environmental dynamics, and the random deployment of WSNs preclude dependence on manual configuration. Inevitably, unattended WSNs must self-organize in response to node failure or changing environmental conditions. The lossy nature of wireless communication caused by the primitive, low-power radio transceivers found in sensor networks poses yet another challenge. Classical routing algorithms are formulated based on a connectivity graph describing which nodes can communicate over a single hop. In a WSN this graph must be dynamically discovered by nodes observing communication events. Connectivity is not a deterministic relation, but a statement of the probability of successful communication. Closely placed nodes may be in communication most of the time and nodes with a higher distance may communicate less reliably, but a few of those may have a strong connectivity. This can be partially attributed to differences in hardware calibration. Hence, an essential building block of WSN is a procedure to determine the neighbors of a node.

The reception of a single message is not an acceptable indicator for a useful link. It is necessary to observe packet success over a longer period of time and to estimate the quality of the link. We have designed and implemented a neighborhood discovery protocol called *Wireless Neighborhood Exploration* (WNX), a modified and extended implementation of TND [10]. WNX determines uni- and bidirectional links and adds a quality descriptor to every link [2]. Based on these bidirectional links, a depth-first spanning tree rooted at a selected node called sink is built. The tree is used to route messages from any node to the sink and vice versa.

The topology discovery algorithm was implemented in an application that was repeated every full hour (see Figure 2). The application is very simple, the nodes merely send their sensor values to the sink in intervals of 10 minutes. At the start of each cycle the neighborhood of each node is determined, this is achieved by running WNX for a specific duration of time. This phase is followed by a pass of the depth-first search computing the routing tree. After this, the sink sends its local time to all nodes in the tree; this is repeated within each data interval. Finally, the nodes start sending their sensor data. The goal of the field trial was to evaluate this procedure using traditional quality of service metrics such as packet loss, effective end-to-end throughput, latency, network connectivity etc.

4.2. Node Hardware

For the experiment, the ESB nodes from the Free University Berlin were used [11]. They consist of the microcontroller 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 transmission power can be tuned in software. Each node has 2 KB RAM and 64 KB EEPROM and is powered by three AA batteries. The sink had a permanent power supply. The power consumption of the nodes according to the specifications varies from 8 μA in sleep mode up to 12 mA when running with all sensors.

4.3. Packaging

The nodes had to be prepared for diverse weather conditions including snow, rain, and sunshine. Waterproof packing was essential. The nodes were shrink-wrapped together with desiccant bags. The foils were then placed in waterproof boxes, which again were put into plastic bags (see Figure 3). The recorded temperature during the experiment varied in a range of almost 40 °C. The results show that the packaging was sufficient to prevent any malfunction of the nodes. The packaging formed an insulation of the nodes and affected the measurements of the sensors. A temperature of 34 °C was recorded by a node of which the box was directly exposed to bright sun. This was accepted, since it was not planned to interpret these measurements.

4.4. Deployment

We deployed 24 nodes on private property of about 140×80 m containing three smaller buildings. The sink and three other nodes were installed inside a building, the others were installed outdoors, spread across the area. In the same building the nodes where prepared. One team member set the real-time clock of each already flashed node and reset counters. Another team member was responsible for waterproof packaging. Two team members installed the outdoor nodes.

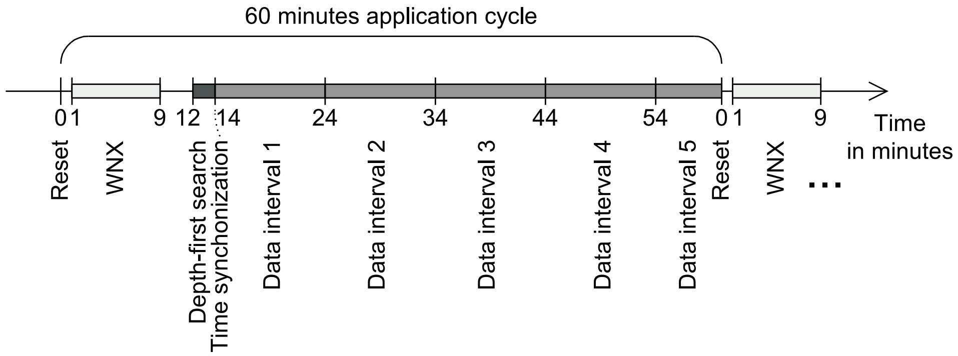


Figure 2. The different phases of the application cycle

Figure 3. Packaging applied for Heathland Experiment

The majority was attached to trees at a height of about 4 meters and some on poles just above ground level. Nodes were fixed with strong adhesive tape. Each position was determined using GPS for the evaluation of the trial.

After the installation all nodes ran the WNX protocol and accepted a few commands. Each node was tested and its transmission power was adjusted to ensure a reasonable routing scenario without nodes having too many neighbors acting as central hubs. The current number of neighbors was determined by querying the neighbor table from each node. Nodes were accessed via radio from nearby with a mobile Tablet PC attached to one node serving as wireless network interface. Frequently more than one attempt was necessary due to communication failures despite the short distance to the node. As a last step each node was set to application mode.

Analyzing data logged during the first application cycle revealed an error in the node software necessitating a re-deployment of a new software version. An analysis of the data showed that nodes having many neighbors frequently reset themselves. The reason was a buffer overflow not discovered in the pre-deployment test phase. As a consequence all nodes had to be collected, unwrapped, re-flashed with the revised software, and re-deployed as described above. The effort was similar to the initial deployment.

4.5. Parameters of Interest

The application of the field trial involved several novel algorithms. These were already tested on a small scale within the lab, their suitability for a long-term outdoor usage can only be tested in a field trial. Among other things the purpose of the trial was to

- find optimal values for various parameters of the communication protocol (e. g. number of retransmissions, size of neighborhood tables),
- compare link qualities as determined by WNX with observed packet success rates over a longer time,
- analyze packet routing using the depth-first tree, and to
- measure variations of clock time and the quality of the time synchronization protocol.

As stated in Section 3.3, the boundary conditions make the storage of a complete trace of the trial almost impossible. It was decided to log data for the analysis of WNX only for selected links, the rationale was that this would give enough insight to optimize internal parameters. To evaluate the mechanism of link quality estimation, the assigned link qualities of all links were logged once per data interval. To compare this data with the channel utilization, each node counted the number of the following events: sent packets, successfully delivered packets, and number of retransmissions. The totals of these numbers were logged once within each 10 minute interval of the application. This allowed to analyze the temporal course of the validity of link estimates. To analyze the routing structures, the depth-first tree was logged once during every application cycle. This allowed to derive properties such as connectivity, hop count, and average branching degree. The neighborhood of every node was logged once per data interval, in combination with the recorded spatial information this gave an insight into the relationship of communication quality and spatial distribution. All records carried a timestamp, this was used to correlate events and compute latencies.

For analyzing the hard- and firmware stability, the number of resets caused by the watchdog and the deviation of the time were logged. The time deviation was accumulated every time a node received a new time packet. If the node was contained within the depth-first search tree, this happened within each 10 minute interval. Furthermore, battery voltage and sensor values were logged periodically.

4.6. Logging Strategy

Our logging strategy had to be essentially wireless, which was mainly dictated by the packaging in use. The EEPROM was too small to store all data, especially since the duration of the trial was unknown. It was only used to store the total number of resets and time deviation information. Plugging cables into the nodes was not possible because they were shrink-wrapped and in boxes. Therefore, the logging had to be wireless. We did not want manual logging either, because the experiment should run without any human attention. To simplify matters, we decided to use a single logging unit, which was the sink node connected to a power-supplied notebook within a building. Hence, wireless automatic logging with routing was used. All nodes within the tree sent their data up the depth-first tree to the sink; the notebook stored the data in files. As a side-effect this traffic was used to evaluate the quality of multi-hop routing. This strategy proved to be only partially successful. The main reasons are:

- Only nodes within the depth-first tree sent data (on the average these trees contained only 50 % of the nodes).
- In one third of the application cycles the depth-first search did not terminate.
- Data sent up the tree got lost due to communication failures (only 50 % of all unicasts were successful).

Nevertheless, a significant amount of data has been logged, which was enough to make a reasonable evaluation.

4.7. Long-term Consideration

To achieve long-term operation of the deployed sensor network, the focus was on application stability and energy conservation. In pre-deployment tests, frequent automatic resets triggered by the watchdog were observed. These resets were caused by non-terminating loops in the firmware due to hard- or firmware problems. It was also observed that nodes sometimes did not receive packets anymore; however, a manual reset solved the problem. This led to the decision to start every application cycle with a software reset (i. e., once every hour). This improved the reliability highly: All but three nodes demonstrably received packets during the whole experiment.

To save energy, the ESB nodes can switch off the transceiver and the sensors. After having built the tree, the leaf nodes disabled their transceivers; they only enabled them for the short periods when sending measurement packets. The inner nodes of the tree disabled their sensors, as they were primarily used for routing. This strategy could have been improved such that nodes that were not in the tree disabled both transceivers and sensors, because they were idle until the start of the next application cycle. Furthermore, during the initial WNX and depth-first search phases, the sensors were not needed and could have been switched off.

Later we discovered that accessing the real-time clock of a node consumes considerable energy. Since the clock was read periodically in an active loop to determine the current application state, this appeared to be another energy bottleneck. A timer-based strategy would have been far less energy-consuming.

5. Conclusion

Field trials are an expensive method to evaluate WSNs and the effort only pays off if the trial is carefully planned. This paper can be regarded as the starting point to define a methodology for planning and monitoring deployments of WSNs. The Heathland Experiment provided valuable insights that would not have been possible with simulations. This enabled us among other things to tune parameters such as packet size, retransmission rate, size of internal data structures etc. To avoid wasting resources through failed or abandoned deployments, more research is needed. The following list contains a selection of issues that need to be addressed:

- packaging that protects the hardware but at the same time does not influence the sensing process,
- distributed calibration of sensors in the field,
- procedures to adopt transceiver parameters individually to local conditions (e. g. signal-to-noise ratio),
- techniques to reprogram the nodes in situ,
- scalable methods for deployments of very large WSNs.

References

[1] G. Pottie and W. Kaiser, “Wireless integrated network sensors,” *CACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[2] V. Turau, M. Witt, and C. Weyer, “Analysis of a Real Multi-hop Sensor Network Deployment: The Heathland Experiment,” in *Proc. INSS 2006*, Chicago, IL, USA, June 2006.

[3] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, “Habitat Monitoring with Sensor Networks,” *CACM*, vol. 47, no. 6, pp. 34–40, June 2004.

[4] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, “An Analysis of a Large Scale Habitat Monitoring Application,” *Proc. SenSys’04*, pp. 214–226, Nov. 2004.

[5] K. Römer and F. Mattern, “The Design Space of Wireless Sensor Networks,” *IEEE Wireless Comm.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[6] A. Woo, T. Tong, and D. Culler, “Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks,” in *Proc. SenSys’03*, Nov. 2003, pp. 14–27.

[7] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, “EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks,” in *Proc. 2004 USENIX Technical Conference*, Boston, MA, USA, June 2004, pp. 283–296.

[8] “Intel Mote,” http://www.intel.com/research/exploratory/motes.htm.

[9] Q. Ye, Y. Zhang, and L. Cheng, “A study on the optimal time synchronization accuracy in wireless sensor networks,” *Computer Networks*, vol. 48, no. 4, pp. 549–566, July 2005.

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

[11] “ScatterWeb,” http://www.scatterweb.net.