Poster Abstract: Reliable and Energy-Efficient Bulk-Data Dissemination in Wireless Sensor Networks

David Gugelmann, Philipp Sommer, Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{gugelmann, sommer, wattenhofer}@tik.ee.ethz.ch

Abstract
Data gathering is one of the most common applications of wireless sensor networks. Such networks are an extremely useful tool for researchers in various domains since they allow for measurements in unaccessible locations, e.g., on mountains, glaciers or in animal habitats. At the same time, the deployment and maintenance of such sites is time consuming and costly. Network reprogramming protocols allow to distribute code updates over the wireless network, which prolongs the interval between human intervention at the deployment site.

While network reprogramming has been studied extensively in the literature, existing bulk data dissemination protocols often do not work well together with data gathering applications which operate at low duty-cycles. We present a reliable data dissemination protocol designed for energy-efficient operation at low duty-cycles. A simple remote procedure call mechanism is used to control and monitor single nodes during the update process, which provides maximal observability during network reprogramming.

1 Introduction
Wireless sensor networks gain momentum as a research tool within different scientific domains, e.g., environmental monitoring. Such sensor deployments strive to gather long-term data with a focus on high measurement accuracy and data yield. Interruptions of the sensing process may have a severe impact on the success of the whole experiment. Therefore, it is indispensable to restrict network downtime to a minimum. Furthermore, human intervention, such as manual reprogramming of sensor nodes, may be costly and even dangerous when sensor nodes are deployed in areas which are difficult to access by humans.

Remote code updating mechanisms, also known as network reprogramming, have been explored in the context of wireless sensor networks. Updating the program image running on a node over the radio typically consists of three steps: (1) announcement of a new image, (2) image dissemination and (3) reprogramming the node.

Deluge [2] employs a "polite-gossiping" algorithm to announce the availability of a new binary image. The data dissemination phase is started when nodes request a new image from their neighbors. Since a single radio message can only hold a limited amount of data, an image has to be split in multiple pages and packets. Deluge employs negative acknowledgments to request missing or corrupted packets again. Finally, an image checksum guarantees the integrity of the image before it is flashed to the program memory. In this work we tackle two major problems of Deluge: the lack of feedback from nodes about the success of network reprogramming and the interference of update packets with other network protocols [3].

2 Data Dissemination Protocol
In this section, we present the basic building blocks of our novel data dissemination protocol for wireless sensor networks. Our work builds upon SlotOS [1], a slotted programming abstraction for TinyOS, which has been designed to ease programming by decoupling different modules of an application. By design, a module is only allowed to be active within its designated time slot. This assumption greatly reduces the code complexity since resource arbitration, such as providing access to the radio chip, is already handled by the operating system. Furthermore, SlotOS aims to reduce idle listening time of the radio transceiver by establishing a global time on all nodes. Each module is scheduled for execution during one or mored dedicated slots within each round of 32 seconds, see Figure 1. The radio transceiver is only powered when requested by the currently running module, otherwise it remains in the power-save mode. The first slot of each round is reserved for the *TimeSync* module, which runs a clock synchronization protocol to align the schedules on all nodes. The *Bidirectional Tree* module builds and maintains a spanning tree rooted at the base station containing only links

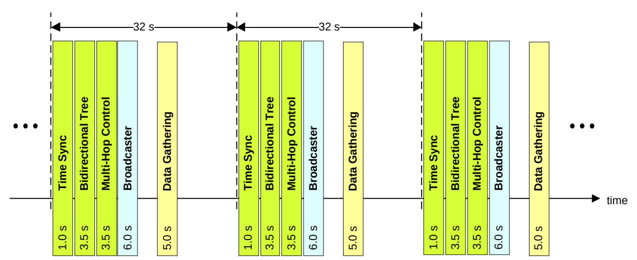

Figure 1. Execution times of modules are scheduled within each round by SlotOS. Modules are allowed to operate within their dedicated time slots only.

SenSys'10, November 3–5, 2010, Zurich, Switzerland.
ACM 978-1-4503-0344-6/10/11

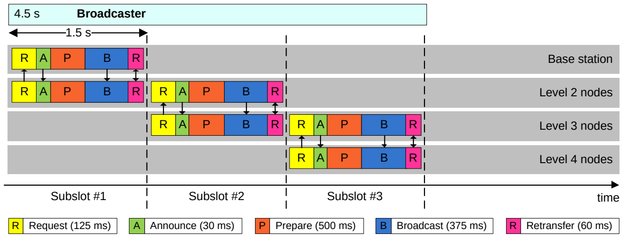

Figure 2. The *Broadcaster* module operates within multiple subslots which are assigned based on the level of a node in the bidirectional tree.

with high bidirectional packet reception rates. This spanning tree is used by the *Multi-Hop Control* module to forward control messages between nodes and the base station. As the next module in the schedule, the *Broadcaster* module is responsible for the data dissemination. It is only active on-demand when there is a new image to be disseminated, otherwise the radio and the microcontroller are kept in sleep mode to reduce the energy consumption. Similar to Deluge, the data object is split into pages each containing 48 packets. One page is attempted to be transfered in each round. The slot for the broadcaster module is further divided into multiple subslots (*Request, Announce, Prepare, Broadcast and Retransfer*), as shown in Figure 2. A node with children in the spanning tree is active during two consecutive subslots. First, it will receive a page from its parent node in the first subslot. Then, it will transmit the page to its children in the next subslot. Consequently, leaf nodes in the spanning tree are only active during one subslot to receive a page. Each subslot is started with a *Request* phase where each child node sends a request for the next page to its parent node. Request packets are sent with a random backoff to avoid collisions. The *Broadcast* phase is reserved to broadcast all data packets corresponding to the current page. To avoid packet collisions by multiple nodes in the same level which simultaneously broadcast data packets, a simple channel reservation mechanism is employed. The child that sends the request for the lowest page number implicitly reserves the following broadcast slot for its parent node. If multiple nodes request the same page number, the node that sent the first request reserves the slot for its parent. The elected parent announces the identifier of the page it will transmit during this round in the *Announce* phase. Next, all nodes power off their radios to save energy while the parent loads the next page from the external flash memory into the RAM (*Prepare* phase). Finally, the *Retransfer* phase provides a small time window to retransfer packets which could not be received correctly. Our experiments revealed that a dedicated retransfer phase at the end of each round reduces the overall latency since it is very likely to have one or a few missing packets in each subslot due to the lossy nature of the wireless channel.

3 Evaluation

In order to demonstrate the feasibility of our approach, we implemented the protocol on the Shockfish TinyNode584 platform, which features a TI MSP430F1611 microcontroller and a Semtech XE1205 radio transceiver. We placed ten nodes in an indoor setup spanning multiple floors. During our test runs, we disseminated a data object of 21886 Bytes, which corresponds to 995 packets in 21 pages. Figure 3 shows the measurement results acquired over 100 repeated test runs and the spanning tree built by the bidirectional tree module. As expected, the radio uptime of a node depends on the number of its children. Node 1 has the highest uptime while the leaf nodes 5,6,9 and 10 have the smallest radio uptime. The dissemination of the complete image took 24.73 rounds on average, which is 17.8% more than the theoretical minimum of 21 rounds. The radio of nodes 2-10 was turned on during 21.7 s on average by the broadcaster module until the image has been completely disseminated. This corresponds to an average data dissemination rate of roughly 1 kB/s, which is about one-ninth of the maximal single-hop data rate of the radio chip.

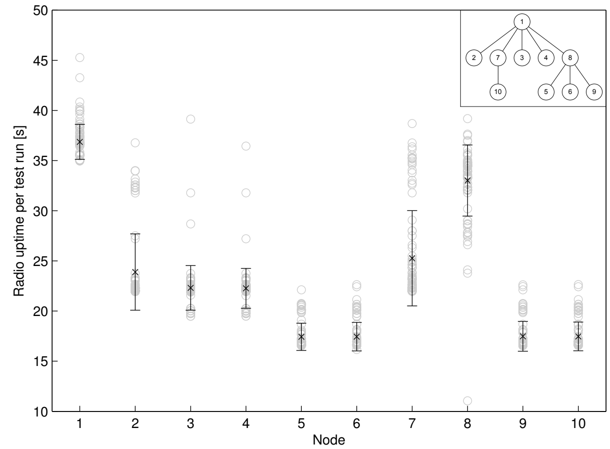

Figure 3. Radio uptime of the *Broadcaster* module during the dissemination of an image of 21 pages. Circles denote the actual measurement results, the bars indicate the standard deviation from the mean value.

4 Conclusion

We have presented a novel data dissemination protocol with a focus on reliability and energy-efficiency. Scheduling image dissemination only during reserved time slots eliminates interference with the regular data gathering protocol and increases the observability during the network reprogramming phase.

5 Acknowledgments

We would like to thank Roland Flury for his support regarding details of the SlotOS implementation. This work was partially supported by the Swiss National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS).

6 References

[1] R. Flury and R. Wattenhofer. Slotted Programming for Sensor Networks. In *IPSN'10*, pages 24–34, 2010.

[2] J. W. Hui and D. Culler. The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. In *SenSys'04*, pages 81–94, 2004.

[3] J. Il Choi, M. A. Kazandjieva, M. Jain, and P. Levis. The Case for a Network Protocol Isolation Layer. In *SenSys'09*, pages 267–280, 2009.