Teaching Wireless Sensor Networks through Testbed Development

Anna Förster
Sensor Lab, Faculty of Informatics
University of Lugano
anna.foerster@ieee.org

Mehdi Jazayeri
Faculty of Informatics
University of Lugano
mehdi.jazayeri@unisi.ch

Abstract
The rapid development of wireless sensor networks (WSNs) and the increasing complexity of the deployed applications force researchers to turn to testbeds to test their communication and application related algorithms and protocols. On the other side, industries’ demand for young professionals with WSN expertise is growing too. These two factors motivated us to develop a highly modular and extensible testbed, called Atelier, as part of our bachelor level course on WSNs. In this paper we share our testbed design and describe its main requirements and challenges.

1 Introduction
WSNs have seen rapid research and industrial development in recent years. Both the costs of individual sensor nodes and their size have been constantly decreasing, opening new opportunities for a wide range of applications: from traditional environmental monitoring to personal and home security, road safety, and space applications. At the same time, Switzerland has been a leader for both research and industry in this domain. The most representative research efforts are the nation-wide competence center for research in wireless sensor networks MICS (http://www.mics.org) and the national program Nanoter (http://www.nanoter.ch). Some widely used WSN hardware platforms are coming from Switzerland too, like the BTNode from ETHZ (Zurich) (http://www.btnode.ethz.ch/) or TinyNode from Shockfish (Lausanne) (http://www.tinynode.com/). The market situation together with our own research efforts in WSNs motivated us to offer a hands-on bachelor-level course on deployment of WSNs. The main objective of this course was to develop and deploy a small size testbed at the University of Lugano. The main design requirements were modularity and extensibility for both research and education. Researchers need to test their communication protocols under various application scenarios and real network settings, and students should be able to implement and integrate small projects.

However, designing such a testbed, especially as part of a bachelor-level course, poses many challenges. The system’s modularity is not only dictated by the usual differentiation of the communication stack into layers and protocols, but also needs to respect fair work load and to enable step-wise integration and deployment. Even complete failure of one of the components should not put the deployment at risk.

In this paper we present the first and successful deployment of Atelier, our system architecture and the implemented communication stack. We emphasize especially its modular architecture and the easy extensibility of its protocols and applications. We believe that our architecture and general design approach can help other researchers prepare and plan their own small size WSN deployments.

2 System overview and architecture
Our main requirement when designing the testbed was its high modularity and extensibility. It needs to support a full sensor network communication stack, including medium access, neighborhood management, routing and application protocols. It needs to support various application scenarios, like regular reporting, data aggregation, event-based reporting and database-like applications. Individual components and interfaces between them need to be designed such that work load is spread evenly among the working groups. Moreover, failure or delivery delay of some of the components must not put at risk the overall testbed deployment.

We chose Scatterweb’s MSB430 platform (http://www.scatterweb.de), as it provides us with a basic and typical sensor hardware, see Figure 1. Our current deployment includes 12 nodes, organized in a regular grid over the 1st floor of the Informatics building at the University of Lugano. There is one base station (a laptop), equipped with a sensor node to be able to communicate to the WSN, and situated in the Sensor Lab at the same floor. For the initial deployment we decided

<table>
<thead>
<tr>
<th>Provider</th>
<th>ScatterWeb, Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>MSP430, 8 MHz</td>
</tr>
<tr>
<td>Memory</td>
<td>5KB RAM, 55KB Flash</td>
</tr>
<tr>
<td>Radio</td>
<td>ChipCon 1020</td>
</tr>
<tr>
<td>OS</td>
<td>ScatterWeb², Contiki</td>
</tr>
</tbody>
</table>

Figure 1. Characteristics of the MSB430 sensor nodes
3 Implementation details

The Atelier architecture is built over the Scatterweb firmware library. This is basically an OS-like library, which provides primitives for sending/receiving messages, reading sensor data, setting software timers, etc. The library does not include any communication protocols or applications. 

3.1 Communication stack

Our OSI-like communication stack consists of a time synchronization component, medium access, neighborhood management and routing layers. It is adapted from the communication stack in [1]. We separate the MAC protocols from the time synchronization, since different independent algorithms exist for both and this modularization enables us to test them separately. The time synchronization protocol is a flooding-based protocol, where the base station regularly sends a time synchronization packet. Each receiving node corrects the time in the packet, synchronizes itself and re-sends the packet. The correction terms are two: an experimentally based term for the duration of sending/receiving the packet through the Scatterweb library, and the processing inside the forwarding node itself. Time stamps are set right before sending / after receiving the packets at the MAC layer to avoid additional delays. Currently we use BMAC [2].

We tested 6 different algorithms for the neighborhood management protocol. The best performing one, with 95% communication reliability, turned out to be a passive listening protocol. It only sends preambles if a neighbor was silent for a very long time and evaluates the quality of neighbors based on their RSSI values and delivery rates. The routing protocol is a greedy geographic routing. Since the network is small and the topology dense, we omitted the implementation of face routing due to memory restrictions. Location information is hard-coded into the nodes, but can be easily added as a separate module in the architecture.

3.2 Applications

Currently we have a regular reporting application, which gathers temperature and humidity data each data interval and sends it to the base station. Further, data aggregation can be turned on, which gathers several data readings from the node itself and from other nodes before sending them together uncompressed in a single data packet to the sink. Additionally, the application layer is designed for the implementation of clustering protocols, data compression protocols, event-based applications and database applications.

Figure 2 shows also a battery monitoring component. This module tracks how many bytes of data are sent/received by the node and estimates how much energy is still available. In the case of alarmingly low battery it notifies the application layer, which in turn may send a dying message to the base station. The node continues to function also after this dying message, but the base station is informed about the situation and battery can be exchanged. The estimation of the battery level is based on experiments.

3.3 Packet generation and consumption

In our WSN OSI model, we allow each component of the communication stack to produce its own packets. Each component is allowed to send them “down” in the stack to be sent out to neighbors, which in turn pass the packets “up” to the same layer as the producer, where they are consumed. Each layer below the producing one is allowed to attach additional information to the packets, while preserving the original contents. For example, neighborhood information like RSSI is gathered while normal data packets are routed.

4 Summary and future work

In this paper we presented our Atelier wireless sensor testbed, designed for both education and research purposes. Its highly modular architecture gives us a strong basis for further development and implementation of cutting edge research on communication problems in WSNs and provides us with a useful and reusable tool for teaching WSNs. Ongoing and future work includes implementation and testing of our MAC level optimizations, routing and clustering approaches. Additionally we are working on extending the testbed and enabling automatic re-configuration, re-programming, and logging of experimental data.

5 Acknowledgment

The work described in this paper is supported by the Daccò foundation under the project Sensori per il futuro: una piattaforma per studi interdisciplinari. We would like to thank especially Amy L. Murphy from FBK-IRST in Italy for her valuable comments. This work would have never been accomplished without the bachelor students and the teaching assistants of the course Atelier V during Fall 2008 at the Faculty of Informatics, University of Lugano.

6 References


Teaching Wireless Sensor Networks through Testbed Development

Anna Förster, Mehdi Jazayeri

Class participants: Adnan Al Hariri, Siegmund Allendorfer, Alessandro Andreani, Andrea Bellani, Christian Caggiano, Paolo Garcioli, Federico Caputo, Andrea David, Alejandro Garcia, Bekim Imeri, Matteo Re Cecconi, Krzysztof Zawadzki, Marco Primi, Andrea Adamoli

Atelier architecture

Currently: periodic reporting, with without packets containing
Planned: event-based notifications, clustering (random and Clique)

Application
Routing
APPL packets
Routing packets
APPL packets
Routing packets
APPL packets
Routing packets
Link beacons
APPL packets
Routing packets
Link beacons
APPL packets
Routing packets
Link beacons
APPL packets
Routing packets
Link beacons
APPL packets
Routing packets
Link beacons

Battery estimator
Time Synchronization
Medium Access Protocol

Battery alerts
Routing
APPL packets
Routing packets
Link beacons

Experience:
- drifts by 1 sec ~ 2-3 days
- experimentally obtained thresholds for sending/receiving bytes

Results: 98% reliability

Currently: flooding-based
Experience: drifts by 1 sec ~ 2-3 days

Atelier’s hardware

Transceiver TI Chipcon CC1020
Clock and radio quartz crystals 32.768 MHz, 14.74 MHz
External mass storage SD-/MM-Card slot
Microcontroller Texas Instruments MSP430 F1612
Acceleration sensor Freescale MMA7260Q
Rel. humidity and temperature sensor Sensirion SHT11

External mass storage SD-/MM-Card slot

Base station

- Monitors the status of each node
- Notifies administrator when battery fails
- Receives and stores sensory data
- Displays sensory data over periods
- Sends parameters to network

Scatterweb MSB430
www.scatterweb.de

Current deployment

Sensor network lab
South (main building)
North (street)

... and the winners are ...
Alessandro and Paolo!

Assignment 3 discussion and midterm recapitulation
Anna Förster
University of Lugano, 22. October 2008

Introduction to Wireless Sensor Networks, Anna Förster
Assignment 3 results for individual nodes

Introduction to WSNs, Anna Förster
Assignment 3

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic

Experience:
- over-pessimistic
- under-pessimistic
- experimentally obtained thresholds for sending/receiving bytes
- over-pessimistic