Electronic Communications of the EASST
Volume 17 (2009)

Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2009 (WowKiVS 2009)

Overcoming a Communication Barrier on the Way Towards a Global Sensor Network

Jochen Furthmüller, Mario Pink, Hannes Hartenstein and Oliver P. Waldhorst

6 pages

Guest Editors: M. Wagner, D. Hogrefe, K. Geihs, K. David
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Overcoming a Communication Barrier on the Way Towards a Global Sensor Network

Jochen Furthmüller, Mario Pink, Hannes Hartenstein and Oliver P. Waldhorst[1]

[1]Universität Karlsruhe

Abstract: In a global sensor network different sensor platforms will be deployed. A grave obstacle on the way of building sensor networks out of different sensor nodes are incompatible implementations of network protocol stacks used with different sensor node platforms. We describe our efforts to overcome this obstacle in a heterogeneous sensor network consisting out of MICAz Motes and Sun SPOTs, both using an IEEE 802.15.4 radio chip. We explain the major differences in the respective network stacks and our approach to bridge them. A network stack that bridges the gap between different platforms allows for more flexible and robust networks.

Keywords: heterogeneous sensor network, network protocol, mac layer

1 Introduction

The vision of a global sensor network (GSN) is inherently tied to the cooperation of different types of sensor nodes. Sensors deployed for different purposes like animal tracking [15], stress monitoring of bridges [2], health monitoring [12] or process automation [16] have to satisfy different requirements.

The need for processing power, communication bandwidth, energy supply, memory, persistent storage and peripheral devices depends on the actual usage scenario. Driven by these diverse requirements a plethora of different wireless sensor nodes has been developed and is available today: MICAz Motes [3], TelosB [4], IMote2 [5], Sun SPOT [17], Scatter Node [14], BTNode [1] to name just a few of them. Beside these commercial products there exist a lot of platforms developed and dedicated for one special application. They all cover a wide spectrum of hardware configurations and operating systems - there is no “one size fits all” in WSN.

Hence, in a GSN different types of sensor nodes must be accessed. One way to accomplish this are dedicated gateways and base stations for each type of sensor. A complementary approach is to harmonize the network protocols of different sensor types and so to increase the compatibility of different sensor platforms. That way sensor networks consisting of diverse platforms can be accessed via the same base stations and even more, heterogeneous sensor networks can be created in which different sensors fulfill different tasks. Being able to use different sensor nodes for different tasks and roles can make a network more flexible and adapted to its environment. However, in order to create a coherent network consisting of different sensor nodes one must have at least a common communication system. Although some of the sensor nodes do employ the same radio chips and do claim to comply with a certain protocol it is a non-trivial task to build a coherent sensor network with off-the-shelf products from different vendors. The ZeuS project [19] aims to employ a prototype of such a heterogeneous WSN. Within in this project the authors encountered a couple of problems concerning the communication between the different node types. This communication gap severely obstructed the development process of the network.

Our contribution is a detailed description of obstacles that hindered us on our way towards a heterogeneous WSN and our measures to overcome them. Although these issues might be specific to the used hardware there are some aspects that can affect building a network with other sensor nodes as well.

In Section 2 we explicate these difficulties concerning communication in heterogeneous sensor networks in general. A more detailed view of the situation and our countermeasures with respect to the used platforms MICAz Motes and Sun SPOTs is presented in Section 3 before we draw our conclusions in Section 4.

2 Communication in Heterogeneous Sensor Networks

In a wireless sensor network communication is an expensive task in terms of energy consumption [8]. As wireless sensor nodes usually have a strictly constrained energy budget, the choice of an adequate communication technology is crucial. Bluetooth and IEEE 802.15.4 [7] are two options that offer a reduced energy consumption compared to 802.11.X for the trade-off of a lower bandwidth. Bluetooth offers an entire network stack whereas IEEE 802.15.4 only specifies the two lowest layers of the ISO/OSI layer model (physical and link layer).

But also on a higher level of abstraction there are a lot of different communication protocols. Multi-hop connectivity can be accomplished by sophisticated routing protocols for ad hoc networks (e. g. AODV [13]), by flooding based protocols (e. g. Trickle, Dissemination [11]) or tree based routing protocols (e. g. Collection Tree Protocol [6]) to name just a few. These different protocols reflect the needs of different use cases of WSN.

A simplified layer model that exhibits the major challenges in creating a heterogeneous sensor network from communications point of view is illustrated in Figure 1. Layer one and two of the ISO/OSI model are combined to a medium access layer, layer three offers routing and forwarding facilities whereas everything above is left for the application.

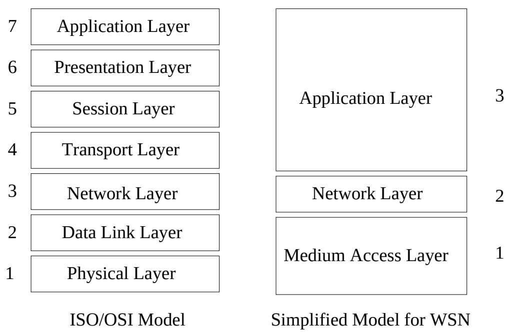

Figure 1: On the left: the ISO/OSI model. On the rigth: A simplified model of communication systems for WSN

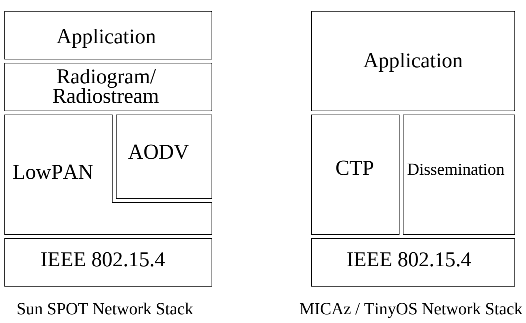

Figure 2: Comparison of network stacks of Sun SPOTs and MICAz Motes / TinyOS

Following this drawing, in order to communicate in a heterogeneous sensor network there are two major obstacles that must be overcome:

- compatible network access: Even using the same or very similar radio chips that comply to certain standards on different sensor nodes does not guarantee for a working communication off-the-shelf across different platforms. Standards might be interpreted unequally or implemented incompletely. During our work we came across several problems, amongst others different addressing schemes, interpretation of header flags and timing issues. We will elaborate on these issues in Section 3.
- consorted network layers: Routing protocols and forwarding policies must be harmonized in order to enable a coherent network structure consisting out of different node types.

3 Harmonization of Network Stacks

The prototype of the ZeuS WSN was implemented with Sun SPOTs and MICAz Motes. The required communication patterns for the communication subsystem of the prototype were propagation of data from a dedicated basestation to one node, a group of nodes or all reachable nodes of the network and propagation of data from a node to a dedicated data sink in the network.

Sun SPOTs and MICAz Motes are to be considered heterogeneous in multiple respects. They differ in memory size, computing power, operating system, programming language and - most significant for interaction - network stacks.

As pointed out in Section 2 we identified two layers that needs adjustments in order to create a seamless heterogeneous sensor network out of Sun SPOTs and MICAz Motes. On the one hand the differences on the medium access layer prevent a direct node to node communication amongst nodes of different types. On the other hand even if this could be accomplished there were no common network protocols available that allow for a multihop communication. Subsequently we describe the issues we came across and the way we coped with them.

We decided to confine code modifications to the Sun SPOT platform as far as possible. That way it was easier to integrate our work within other subprojects of the ZeuS project: only one set of libraries and applications needs to be modified in order to use the harmonized network stack.

3.1 Medium Access Layer: Addressing Schemes

The IEEE 802.15.4 standard allows two ways of addressing source and destination inside a packet. Actually each node in a IEEE 802.15.4 network is supposed to have a 64 bit long address. As IEEE 802.15.4 packets are are rather small there is a second addressing scheme defined, using 16 bit long addresses.

Each Sun SPOT is assigned a 64 bit address. A Sun SPOT's radio chip is configured to accept all packets bearing this address or the broadcast address. Packets with other addresses are to be discarded. By contrast MICAz Motes in combination with TinyOS use 16 bit addresses configured by the application programmer by default. Hence, in order to establish connectivity on the medium access layer, the addressing scheme needs to be harmonized. We decided to modify the Sun SPOT software, as we regard it desirable to save space in data packets. The changes were designed in a way that once a Sun SPOT was using the modified libraries its addressing scheme could be changed at boot time. By setting a user defined property `ADDRESS_MODE=SHORT` a Sun SPOT is configured to use a 16 bit address.

The most relevant modifications took place in the following classes of the Sun SPOT libraries:

- `com.sun.spot.peripheral.SPOT`: Checking address mode, defining the own address
- `com.sun.spot.peripheral.radio.CC2420`: Configuring radio chip with 16 bit address, enabling address recognition
- `com.sun.spot.peripheral.radio.MACBase/-MAC.Layer`: Initializing MAC layer with corresponding address
- `com.sun.spot.peripheral.radio.RadioPacket`: Initializing data structures with the right offsets for packets with short addresses
- `com.sun.spot.peripheral.radio.MicazPacketDispatcher`: Offering access to changed functionality. Essentially derived from the classes RadioPacketDispatcher and LowPanPacketDispatcher.

3.2 Medium Access Layer: Timing Issues

After having established connectivity between Sun SPOTs and MICAz Motes on the medium access layer we observed an interesting phenomenon. For packets sent from a Sun SPOT to a MICAz Mote the corresponding acknowledgment sometimes arrives too late. This results in a `NoAckException` thrown by the Sun SPOT's MAC layer. This makes the sending application believe that the packet was not received properly. Depending on the implemented policy a retransmission is scheduled. Fine-grained measurements to determine whether the Sun SPOTs do not wait long enough or the MICAz Motes send the acknowledgement too late are still to be done. As a workaround for this problem we extended the time slot that a Sun SPOT is waiting for an acknowledgment before it considers a packet as lost. It should be mentioned that this can have a negative side effect: The mapping between data packets and corresponding acknowledgment in IEEE 802.15.4 is done only by a 8 bit sequence number. A sending node considers only acknowledgments that arrive in a very short time slot after the sending operation. This reduces the probability that another station is acknowledging another packet that carries the same sequence number incidentally. Prolonging this time slot can result in an increased rate of wrongly acknowledged packets. However, we still decided to do it like that as we did see no easy way to accelerate the acknowledging process of the MICAz Motes.

3.3 Medium Access Layer: Header Flags

One of the control bits in an IEEE 802.15.4 packet header denotes whether a packet contains one or two fields for PAN (Personal Area Network) identifiers. If only one PAN id is specified, it is assumed that source and destination PAN is identical. As Sun SPOTs did not support the concept of PAN and PAN coordinator, it was not expected to receive a packet with different PAN ids. We disabled the checking of this header field so that a Sun SPOT can accept packets that are formatted as a packet containing source and destination PAN ids.

3.4 Network Layer: Multihop Connectivity

MICAz Motes and Sun SPOT's use different network protocols to provide multi-hop connectivity: While TinyOS comes along with implementations of CTP and Dissemination, Sun SPOTs feature a combination of LowPAN and AODV as network protocols (see Figure 2). CTP can be classified as a proactive routing protocol that builds a routing tree towards a predefined destination whereas AODV is a reactive routing protocol that allows flexible communication with multiple communication counterparts. Dissemination in turn is a flooding based protocol that tries to reduce the communication overhead and still provide a high probability of reaching all nodes. These protocols have been designed for different purposes. For the ZeuS project the decision was taken to complement the Sun SPOTs' network stack with an implementation of CTP and Dissemination.

3.5 Application Layer: Payload Length

Another significant difference between the two platforms is the maximum payload length. TinyOS incorporates the concept of the TinyOS message. The according data structure message_t uses fixed offsets for header data ignoring their actual offset in the radio packet that is to be sent. That way the data that is given from an upper layer to a lower layer does not need to be copied into a new data structure several times. This saves time and energy, however the price that has to be paid is a reduced maximum payload of 29 bytes (see [10]). Sun SPOTs do copy the payload data into a IEEE 802.15.4 packet. This means that they can send a payload longer than 29 bytes, the actual size depends on the length of the headers (multi-hop or single-hop header, 64 bit addresses or 16 bit addresses and others). This means that an application that is supposed to run on this heterogeneous network out of Sun SPOTs and MICAz Motes should confine itself to sending packets with no more than 29 bytes of payload, each. We decided to adopt the maximum length of the MICAz / TinyOS system for the Sun SPOTs. That way it is possible to use off-the-shelf MICAz sensor nodes in the heterogeneous network to be built.

4 Conclusion

For the creation of a GSN, communication across the border of different sensor platforms is a fundamental condition. But also single deployments can profit from applying different platforms for different roles. In the ZeuS project a prototype of a heterogeneous sensor network is being developed. A serious obstacle on the way to seamless communication in a heterogeneous environment are the different network stacks available for different nodes, even if they employ the same radio chips. This work describes our experiences with Sun SPOTs and MICAz Motes within the ZeuS project. We identified 5 major problems and described our approaches to overcome them. The necessary modifications in the Sun SPOTs' library were published on java.net. They have been picked up by the development team and included into the ongoing development of the Sun SPOT project. At the time of writing the beta version of the latest Sun SPOT development kit release (called blue) supports 16 bit addresses as well.

With further investigations in the area of heterogeneous sensor networks we hope to further alleviate the usage of diverse sensor platforms within one single sensor network deployment.

Bibliography

[1] BTNode Website. http://www.btnode.ethz.ch/

[2] K. Chebrolu et al.. BriMon: A Sensor Network System for Railway Bridge Monitoring In Proceedings of the Sixth International Conference on Mobile Systems, Applications, and Services (MobiSys), 2008.

[3] Crossbow. Micaz datasheet, retrieved from http://www.xbow.com.

[4] Crossbow. TelosB datasheet retrieved from http://www.xbow.com..

[5] Crossbow. Imote2 datasheet retrieved from http://www.xbow.com..

[6] R. Fonseca et al.. TEP 123: Collection Tree Protocol (CTP), retrieved from http://www.tinyos.net, 2007.

[7] IEEE 802.15.4, retrieved from http://standards.ieee.org/getieee802/802.15.html, 2003.

[8] O. Landsiedel et al.. Accurate Prediction of Power Consumption in Sensor Networks, 2005 The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.

[9] P. Levis and G. Tolle. TEP 118: Dissemination, retrieved from http://www.tinyos.net, 2007.

[10] P. Levis. TEP 111: message_t, retrieved from http://www.tinyos.net, 2006.

[11] P. Levis et al.. Trickle: A Self-Regulating Algorithm for Code Maintenance and Propagation in Wireless Sensor Networks. In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI), 2004.

[12] J. Paek et al.. A Wireless Sensor Network for Structural Health Monitoring: Performance and Experience The Second IEEE Workshop on Embedded Networked Sensors, 2005

[13] C. Perkins et al.. Ad hoc On-Demand Distance Vector (AODV) Routing, 2003.

[14] Scatter Website. http://www.scatterweb.com.

[15] P. Sikka et al.. Wireless sensor devices for animal tracking and control 29th Annual IEEE International Conference on Local Computer Networks, 2004.

[16] L. M. Sá de Souza SOCRADES: A Web Service Based Shop Floor Integration Infrastructure The Internet of Things, page 50-67, 2008

[17] Sun Microsystems. Sun SPOT developer's guide, retrieved from http://www.sunspotworld.com, 2007.

[18] Texas Instruments. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, 2006.

[19] ZeuS Project Website. http://www.zeus-bw-fit.de