A Cross-Layer Protocol Evaluation Framework on ESB Nodes

[Demo Abstract]

Yves Igor Jerschow Björn Scheuermann Christian Lochert Martin Mauve

Institute of Computer Science
Heinrich Heine University Düsseldorf
Düsseldorf, Germany

yves.jerschow@uni-duesseldorf.de
{scheuermann, lochert, mauve}@cs.uni-duesseldorf.de

ABSTRACT

Due to the fact that the MAC layer of commodity 802.11 wireless network devices is usually realized in a proprietary firmware module, it is difficult to conduct real world evaluations of novel MANET cross-layer protocols that require a modified MAC layer. We demonstrate a testbed framework based on ESB sensor nodes. Due to their open firmware these nodes allow for the implementation of arbitrary MAC modifications and cross-layer interactions. This provides an opportunity to test MANET and Mesh network protocols with this kind of modifications in the real world on commercially available hardware. Our framework contains utilities and modules which automatically detect the network topology for a given node placement, construct and deploy static routing tables along predefined paths and allow an analysis of the network traffic by logging the packet transmissions. The demo points out the feasibility of experiments and cross-layer implementations with our framework on the ESB nodes.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Protocol verification; C.2.1 [**Network Architecture and Design**]: Wireless communication; C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Experimentation, Measurements, Verification, Performance

Keywords

wireless networks, ad hoc networks, experimentation, real-world implementation, cross-layer, framework

REALMAN'06, May 26, 2006, Florence, Italy.
ACM 1-59593-360-3/06/0005.

1. INTRODUCTION

Real world experiments should be a major component of any MANET protocol evaluation. They are required to evaluate the impact of external factors that are not modeled appropriately in simulators. This poses a significant challenge for cross-layer protocols, since these protocols often need to use information from, or to change the behavior of the MAC. However, real network devices are mostly equipped with an IEEE 802.11 wireless LAN adapter. Due to the proprietary firmware, real world experiments with MAC modifications are very hard to conduct.

We propose to use a different platform than IEEE 802.11—the ESB sensor nodes (Figure 1)—to overcome this problem. These devices have been developed for wireless sensor networks at the Freie Universität Berlin in the ScatterWeb project [2] and are commercially available. The functionality of these nodes ranges from a motion detector to a radio communication device. They have been used in sensor network testbeds, e. g. in [1, 5, 6]. For our purposes, the most interesting feature, however, is the open-source firmware, because it is appropriate for cross-layer protocol implementations. It is possible to manipulate every part of the software. In particular, one can control every single bit transmitted on the wireless medium. Consequentially, these devices allow an implementation of nearly all modifications a network researcher might want to experiment with, e. g., changes on the backoff mechanism, arbitrary cross-layer callbacks and so on.

The demo accompanies our paper [4], where we present an ESB-based framework to conduct real-world experiments more easily and conveniently. Within the standard ESB firmware it is not possible to store data or to log events during an experiment. Hence it is also not possible to gather the desired performance data or observe the behavior of the protocol. Our implemented framework allows to gather the information. Furthermore, the nodes are now able to support the experiment conductor, e. g., by providing information about the topology they form. Thereby the ESB sensor network platform is transformed into a cross-layer testbed platform for MANET protocols.

2. THE FRAMEWORK

Our framework provides the "infrastructure" that is necessary to implement and assess cross-layer protocols in the real world, on the ESB sensor node platform. The demo shows some interesting parts

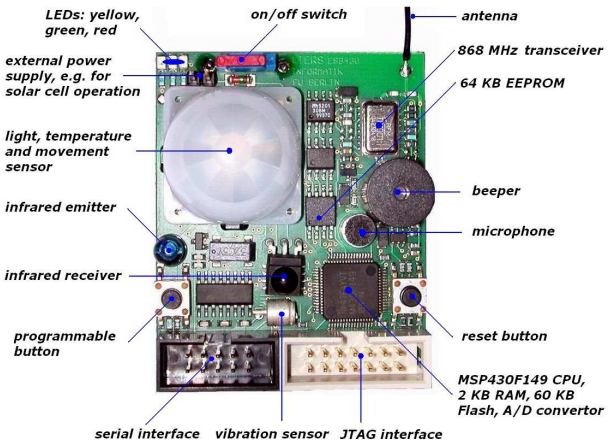

Figure 1: An ESB sensor node.

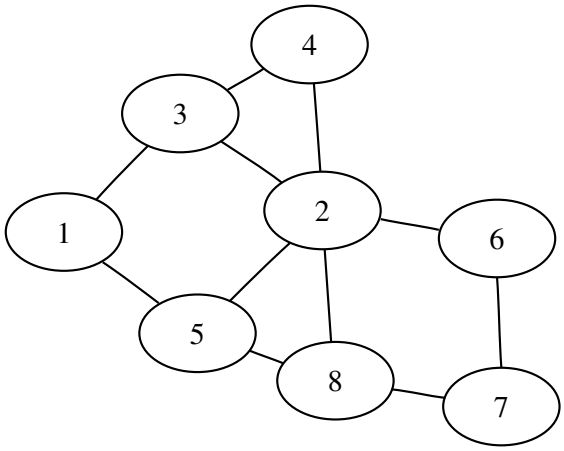

Figure 2: Discovered network topology graph visualized by GraphViz.

of the software, particularly the parts that are used during experiments. Before describing the demo itself, we give a short overview of the parts of our framework that are relevant for the intended presentation.

The main focus of our current work is on MAC/transport layer interactions. Therefore we were looking for a way to eliminate routing protocol effects from our experiments. The intention is to be able to distinguish routing protocol influences from inherent MAC and transport layer issues. Our solution to this challenge is to use a static topology with static routing. To support this on the ESB platform, we have implemented a static multihop routing module, including an appropriate handling of routing tables and a user-friendly tool chain for their creation.

New routing tables can be deployed automatically in the network. Since in experiments the used topology might change quite often, this is a common task that needs to be made as easy as possible. Therefore, a mechanism has been implemented that uses the routes given in the set of routing tables themselves to supply each node with its routing table. Thereby new routing tables can be deployed while leaving the nodes in-place and without visiting each node individually.

When an experiment is set up it is usually hard, but necessary, to verify that the actual topology matches the intended one. The topology exploration module of our framework is able to gather topology information and deliver it to the user. The information provided by the topology exploration makes this verification easier. Additionally, it can be used for the creation of routing tables, either manually or automatically.

A token-based depth-first search strategy is employed by the topology exploration algorithm. The ESB nodes show the state of the distributed algorithm in each node by LEDs. Therefore the progress of the topology exploration can be observed while the algorithm is running.

Finally, a last module of our framework provides a logging facility. The packets received, sent and overheard in all the nodes can be logged, down to single packet (re)transmission attempts. Detailed logging is crucial for both debugging purposes and for a well-founded result analysis. Logging for the ESB nodes is a challenging task especially because of the limited resources: there is only very limited storage space available for log entries. We use a highly compressed bit-field format to store the logs in the nodes' EEPROM.

After conduction of an experiment, a helper application transfers the log data to a desktop computer and creates a logfile. The format of this file is text-based, and it is thus much easier both to read and to parse (e. g., for performance analysis) than the binary data in the EEPROM of the ESB nodes.

3. THE DEMO

For the demo network a number of ESB sensor nodes with our modified firmware is used. Some of the nodes are connected to notebook PCs. Figure 3 shows such a setup. A terminal program running on the notebooks is used to interact with the firmware. By using the notebooks it is, for example, possible to trigger the transmission of data packets and to get live feedback on all packet transmission and reception events, down to single MAC layer frames.

In addition to the terminal program for direct interaction with the ESB firmware, the supporting tools from our framework are installed on the notebooks. The topology exploration service can explore the topology of the demo network. An interface to the graph drawing toolkit graphViz [3] generates graphical representations of the network topology. Figure 2 shows an example network topology graph generated with these tools. This graph can be displayed on one of the notebooks. By configuring different transmission power rates it is possible to create topologies where not every node is able to reach every other node directly. However, this depends to a certain extent on the environmental conditions.

After the exploration of the topology the routing table distribution service deploys the routing setup. One can use either automatically calculated optimal paths or a "virtual" topology configured in human-readable routing table files. It is demonstrated how this can be used to change the routing paths used by the nodes. Some routing table definitions for different topologies are prepared.

LEDs on the sensor nodes blink when a packet (e. g., an echo request) is forwarded, providing immediate visual feedback on the route taken by the packet. The green LED signalizes that a packet is generated, yellow LEDs show that packets are forwared, and the red LED signalizes the reception of a packet. This augments the feedback provided by the terminal windows on notebooks. Finally, the logging service can read event logs from the nodes and convert them to a simple text-based format. This is demonstrated in order to show that it is easy to obtain a solid information basis for performance evaluations.

Figure 3: Demo network setup.

4. REFERENCES

[1] A. Dunkels, L. M. Feeney, B. Grönvall, and T. Voigt. An integrated approach to developing sensor network solutions. In *Proceedings of the Second International Workshop on Sensor and Actor Network Protocols and Applications*, August 2004.

[2] Freie Universität Berlin, Computer Systems Telematics. ScatterWeb Project. http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb_net.

[3] Graphviz—graph visualization software. http://www.graphviz.org.

[4] Y. I. Jerschow, B. Scheuermann, C. Lochert, and M. Mauve. A Real-World Framework to Evaluate Cross-Layer Protocols for Wireless Multihop Networks. In *Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN 2006)*, May 2006.

[5] J. Schiller, A. Liers, H. Ritter, R. Winter, and T. Voigt. ScatterWeb – Low Power Sensor Nodes and Energy Aware Routing. In *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS 2005)*, January 2005.

[6] S. Schmidt, H. Krahn, S. Fischer, and D. Wätjen. A Security Architecture for Mobile Wireless Sensor Networks. In *First European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, August 2004.