

Demo Abstract: Bridging the Gap between Simulated Sensor Nodes and the Real World

Tobias Baumgartner¹, Daniel Bimschas², Sándor Fekete¹, Stefan Fischer²,
Alexander Kröller¹, Max Pagel¹, and Dennis Pfisterer²

¹ Braunschweig Institute of Technology, IBR, Algorithms Group, Germany

² Institute of Telematics, University of Lübeck, Germany

{t.baumgartner,s.fekete,a.kroeller,m.pagel}@tu-bs.de,
{bimschas,fischer,pfisterer}@itm.uni-luebeck.de

Abstract. We present an architecture for the interconnection of simulated sensor nodes and real node hardware. The simulator is therefore running in real-time, and the simulated nodes are able to exchange messages with real sensor nodes as if they were sent over the radio. This runs fully transparent for the application—and is well suitable for debugging purposes and general algorithm development. It is even possible to use exactly the same algorithm implementation for both simulated nodes and real sensors.

Keywords: Sensor Networks, Simulation, Testbeds, Virtual Links.

1 Introduction

Algorithm development for wireless sensor networks (WSNs) is still a challenging task. It involves embedded programming on tiny micro-controllers with well-known problems such as alignment issues, unpredictability of interrupt service routines (ISRs), and a general lack of debugging possibilities. Furthermore, algorithms are mostly distributed, and thus potential errors may only occur in specific situations and are unreproducible due to oscillator variances on the nodes or message loss. A common approach is to run algorithms in simulators before testing on a real testbed. However, results from simulation are often not comparable with real-world experiments, especially when different implementations are used for the simulated nodes and real sensor nodes.

Different solutions have been presented over the past years to obviate these problems. With TOSSIM [5], it is possible to run the same code in a simulator and on real nodes. A similar approach has been presented by Wittenburg and Schiller [9], who extended the ns-2 simulator to run ScatterWeb applications—again, without changing any line of application code. With COOJA [6], it is possible to run Contiki applications before flashing them onto a sensor node. The common denominator of these approaches is that code is either run in a simulator or on hardware—without having a link between simulation and experiment, and thus without appropriate debugging possibilities when an error occurs in the testbed. To overcome these drawbacks Österlind et al. [7] presented an approach

with sensor network checkpointing. The state of all nodes in a testbed can be saved, and put into a simulator for further debugging. In addition, the nodes' states can also be transferred from the simulator back to the testbed.

In our demonstration, we go one step further. We enable simulated nodes to directly communicate with real sensor nodes, whereby it is possible to run the same application code in the simulator and on the nodes. The architecture is based on so called virtual links [1,3], which enable the connection between two nodes that are not in direct communication range. Messages that are sent by a node are automatically passed to a gateway, which in turn injects the message in the simulator, where it is received by a simulated node. The other way around, messages can also be sent from the simulator to a real node over the same connection.

The technique allows for embedding real sensor nodes into arbitrary topologies. Nodes can be placed at critical sections in the network, e.g. to evaluate the behavior of real nodes when being the bottleneck of a complex algorithm that is run in a large-scale simulation.

In Section 2, the architecture of the system, which allows to connect simulated nodes to real sensor nodes is presented. Section 3 describes the mode of operation of our demonstration.

2 System Architecture

The overall architecture has been developed in the context of the EU-project WISEBED [8], which aims at the interconnection of multiple testbeds scattered around Europe. These interconnected testbeds can be configured to behave like a single testbed, hiding the actual distribution to the sensor node application. This is achieved by establishing the aforementioned virtual links between individual nodes. In addition to linking real testbeds, a simulator can be integrated into the testbed—with virtual links between simulated and real nodes, see Fig. 1(a).

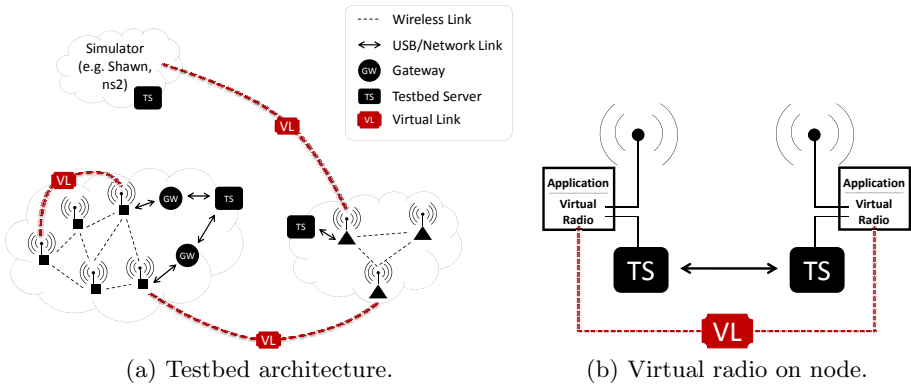


Fig. 1. Wisedbed Architecture for Virtual Links

Application code can be developed using the Wiselib [2], a platform independent C++ algorithm library for heterogeneous sensor networks. Applications using the Wiselib can be compiled for several platforms such as iSense, Contiki, or even simulators. Furthermore, the Wiselib comes with built-in support for virtual links.

Virtual Links. With virtual links, one can connect different sensor node testbeds, appearing to the user as only one large testbed. Nodes in the different testbeds are linked to each other, behaving as if they were in direct communication range. A virtual link transparently tunnels messages between these nodes through the Internet, injecting the message at the destination node as if it was received via the radio. As shown in Fig. 1(b), we use a virtual radio that in turn uses both the real hardware radio and a serial connection to a gateway to be able to send/receive either via radio or a gateway. This happens transparently to the application, depending on the current set of virtual links that are configured for the individual node.

Simulation Environment. We extended the simulator Shawn [4] to be integrated in a virtual topology. First, we added the ability for real-time simulation, so that applications are executed at the same speed as on real sensor nodes. Second, basic multi-threading capability was added to be able to inject messages from real nodes at run-time. Finally, the Wisebed Web Service API was implemented to allow the connection to a real testbed—this way, nodes are not aware whether they communicate with a simulated node or with a real one.

Testbed Access. For the interconnection of a testbed with either other testbeds or a simulator, we developed a Java-based gateway software, which connects itself to the nodes inside the testbed and exposes them via the Wisebed Web Service APIs to the outside. These APIs provide a set of operations that allow researchers to manage the testbed nodes and run experiments. These include but are not limited to re-programming the nodes, collecting debug messages and sending commands to them. In addition, we added virtual link support, so that messages can be received via the Web Service API, or sent to other testbed instances via the Internet. In order to allow higher message rates than possible using the Web Service interface, we furthermore integrated a much more efficient message-based interface that uses direct TCP socket connections.

3 Demonstration

In our demonstration, we present a connection between the Shawn simulator and real sensor nodes. We use the visualization module of Shawn, where nodes can be drawn at run-time, providing a live representation according to their current state.

We have also three types of nodes: Sensors, actuators, and bridges. The sensor and a bridge node are linked to one Shawn instance, the same bridge node and the actuator to another Shawn instance. Whenever the sensor—a light sensor—detects an event (darkness or lightness), it sends a message to the actuator. The message is thereby routed through the first Shawn instance to the bridge node,

which passes it to the second Shawn instance. From there, it is routed to the actuator—a light, which is turned on or off.

Acknowledgement. This work has been partially supported by the European Union under contract number ICT-2008-224460 (WISEBED).

References

1. Baumgartner, T., Chatzigiannakis, I., Danckwardt, M., Koninis, C., Kröller, A., Mylonas, G., Pfisterer, D., Porter, B.: In: Silva, J.S., Krishnamachari, B., Boavida, F.L. (eds.) EWSN 2010 LNCS, vol. 5970, pp. 210–223. Springer, Heidelberg (2010)
2. Baumgartner, T., Chatzigiannakis, I., Fekete, S.P., Koninis, C., Kröller, A., Pyrgelis, A.: Wiselib: A generic algorithm library for heterogeneous sensor networks. In: Silva, J.S., Krishnamachari, B., Boavida, F. L.(eds.) EWSN 2010 LNCS, vol. 5970, pp. 162–177. Springer, Heidelberg (2010)
3. Bimschas, D., Danckwardt, M., Pfisterer, D., Fischer, S., Baumgartner, T., Fekete, S.P., Kröller, A.: Topology virtualization for wireless sensor network testbeds. In: Proceedings of the 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2010), Berlin, Germany. ICST (May 2010)
4. Kröller, A., Pfisterer, D., Buschmann, C., Fekete, S.P., Fischer, S.: Shawn: A new approach to simulating wireless sensor networks. In: Proceedings of the 3rd Symposium on Design, Analysis, and Simulation of Distributed Systems (DASD 2005), pp. 117–124 (2005)
5. Levis, P., Lee, N., Welsh, M., Culler, D.: Tossim: accurate and scalable simulation of entire tinyos applications. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003, pp. 126–137. ACM, New York (2003)
6. Österlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T.: Cross-level sensor network simulation with cooja. In: Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006), Tampa, Florida, USA (November 2006)
7. Österlind, F., Dunkels, A., Voigt, T., Tsiftes, N., Eriksson, J., Finne, N.: Sensor-net checkpointing: Enabling repeatability in testbeds and realism in simulations. In: Roedig, U., Sreenan, C.J. (eds.) EWSN 2009. LNCS, vol. 5432, pp. 343–357. Springer, Heidelberg (2009)
8. Seventh Framework Programme FP7 - Information and Communication Technologies. Wireless Sensor Networks Testbed Project (WISEBED), ongoing project since (June 2008), <http://www.wisebed.eu>
9. Wittenburg, G., Schiller, J.: Running real-world software on simulated wireless sensor nodes. In: Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2006), Uppsala, Sweden, pp. 7–11 (June 2006)