RealSim: Real-time Mapping of Real World Sensor Deployments into Simulation Scenarios

Moritz Strübe*, Simon Böhm*, Rüdiger Kapitza*, Falko Dressler†
*Dept. of Computer Science, University of Erlangen, Germany
†Institute of Computer Science, University of Innsbruck, Austria
{struebe,boehm,kapitza}@cs.fau.de, falko.dressler@uibk.ac.at

ABSTRACT

We present an approach to improve simulation results by automatically mapping the network topology of real wireless networks into a simulation environment. We basically target two problem domains: First, simulations are often used with rather arbitrary configurations, i.e., the used settings may strongly deviate from realistic setups. Secondly, and as our key motivation, we aim to use the simulator to experiment with different algorithms and settings in parallel to an existing sensor network installation. That means that we want to exactly rebuild the real network in order to estimate the performance of our new settings before deploying them into the network. In our demo, we show that not only the physical environment influences the propagation of radio waves, but also the hardware. These changes are incorporated in real-time into our simulator. Even though the reasons for the strong variations are out of the scope, the consequences for the sensor network are significant.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.4 [**Performance of Systems**]: *Performance attributes*

General Terms

Experimentation

1. INTRODUCTION

Simulating algorithms and protocols for Wireless Sensor Networks (WSNs) has many advantages. One is of course that debugging code as well as making measurements can be be done with less effort. By running parallel instances, parameters can be tuned or automated testes can be run to make sure that nodes must not be manually recovered after flashing a flawed firmware. Nonetheless, there is typically a huge gap between simulation results and the real world. While most parts of the hardware can be emulated with very high accuracy using emulators like MSPsim[1] or Avrora[2], simulating the radio communication accurately is extremely challenging [1]. We argue that predicting the right simulation parameters without actual measurements is out of question: In fact, during our experiments we were able to witness significant changes of link quality by just moving some sensor node a few centimeters.

[1]`http://www.sics.se/project/mspsim`
[2]`http://compilers.cs.ucla.edu/avrora/`

WiNTECH'11, September 19, 2011, Las Vegas, Nevada, USA.
ACM 978-1-4503-0867-0/11/09.

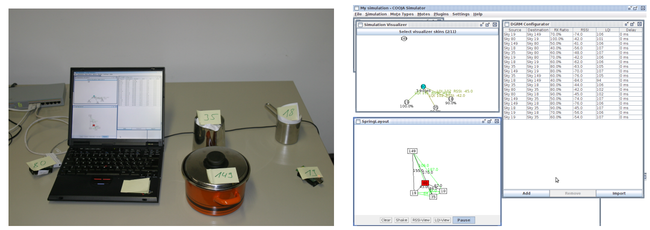

Figure 1: Demo description: We monitor the wireless network conditions (left) and reproduce them in real-time in the simulator (right)

These observations show that precise knowledge is needed to make realistic predictions about how software may behave in the real network. In this demo, we present the RealSim system, which provides real-time mapping of real world sensor deployments into simulation scenarios. We extended the COOJA network simulator to parameterize the wireless communication using live monitoring data, such as packet loss, Link Quality Indicator (LQI), and Received Signal Strength Indicator (RSSI) values, from a real network. Our extension also adds and removes nodes automatically during the simulation. In order to replay the measurements we record the monitored data, thus allowing to reproduce different configurations. Figure 1 shows the demo setup. We show how the conditions of the wireless network are monitored and reproduced in the simulator in real-time.

2. IMPLEMENTATION

In order to obtain some first results, we implemented a simple monitoring application. We basically broadcast at a certain period, e.g 1 s, whereby each broadcast is sequentially numbered. This way, the receiver can easily detect lost packets. Packet loss, the LQI and RSSI values measured by the local sensor hardware are aggregated and sent to the monitoring sink, where the information is collected by a client application via a serial connection.

The collected information is then forwarded to the COOJA simulator using TCP/IP, making it location independent. COOJA has been extended using a plug-in that updates the currently running simulation: It adds the newly detected

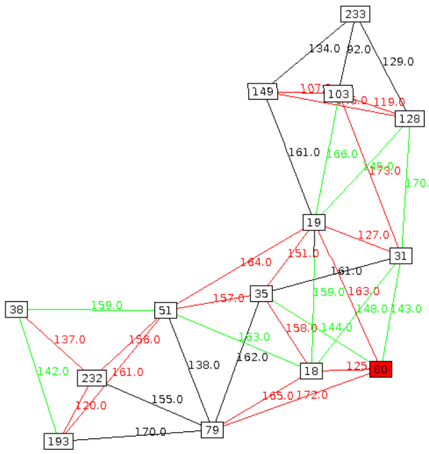

Figure 2: Network topology as calculated according to the monitored wireless conditions in a sensor network spreading most offices in our institute

nodes and updates the connections of the Directed Graph Radio Medium (DGRM) used in the simulation.

Currently Cooja evaluates only the packet loss measure. We plan to also incorporate LQI and RSSI, but currently, we use these values to draw a connection graph using a mass spring model. This basically replaces connections with springs, where the length of the spring is represented by the the LQI and RSSI values.

Figure 2 shows a much larger example of the resulting network topology. We deployed sensor nodes in the offices in our institute. RealSim was used to continuously monitor the conditions in the wireless network. The resulting topology as reproduced in the simulator roughly resembles the physical structure of the floor (it is U-shaped). Although the nodes 103, 149 and 233 were placed only a hand-width apart, they were pushed apart in the layout due to node 103 giving bad RSSI values for near nodes.

3. FINDINGS AND DISCUSSION

We are aware that our measurements and simulations done using Cooja are currently only a very vague resemblance of the real network. To allow better predictions we need to improve the link quality estimation, as well extend Cooja to take these values into account. None the less we believe that these results can give a better predictions about the behavior of an application in a real network, then is possible doing so manually. Especially as the asymmetric links, which may have significant influence on routing are represented.

Also the monitoring application is the only application and it is collecting data with an high frequency. While this might be suitable for a shot term, data should also be collected while the network is working. Thus allowing to monitor the network, but also have accurate data to adjust parameters or ensure that a software update will be seamlessly.

The automatic placement of nodes, respectively their distance, does not influence the simulation itself, but is considered a nice eye-candy. It does allow to get a “feeling” for the network. Especially as the network layout my strongly

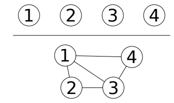

Figure 3: Deviation from real topologies: node 1 has best connectivity, thus, the physical topology (top) is different to the connectivity graph (bottom)

differ form the physical one. Figure 3 outlines this issue: While all nodes are physically in a straight line and nodes `2, 3, 4` receive just their neighboring nodes, node `1` has good reception and pulls all nodes toward itself.

We are currently working on an extended version of the system with optimized monitoring capabilities, i.e., more intelligent aggregation techniques as well as improving the results of the simulation. We are also about to support heterogeneous nodes that may have different radio chips and antennas.

4. RELATED WORK

In the literature, it has been reported that getting realistic simulation results for WSNs is a non-trivial task [1]. Most effort on the accuracy of physical layer, i.e., the link-behavior, has been spend in extending network simulators like NS-2 or OMNeT++ [2]. However, such simulators are not able to fully emulate the actual hardware, but provide abstract models of the underlying hardware and communication links. Applications need therefore to be ported or, in very rare cases, at least be re-compiled to run in theses simulators. While the behavior of a network component, e.g., the MAC protocol, is likely to be similar, this is not the case for a whole application. Complex calculations on resource-limited micro-controllers may delay the handling of packets and influence the behavior of the network.

Österlind *et al.* presented an approach that is orthogonal to our concept of mapping the environment of the sensor network into the simulator. They implemented a check pointing mechanism that allows to stop the whole system, saving its state and rolling it back [3]. This not only support to repeat testes starting in a certain state, but also to move the state to and from the simulation. Even though this allows to analyze the internal state in detail, it is very unlikely that the simulated network will behave similar to the real one.

With our RealSim system, we improve the simulation model by combining real nodes with simulated ones. We implemented this system using the COOJA simulator and TelosB sensor motes.

5. REFERENCES

[1] G. P. Halkes and K. G. Langendoen. Experimental evaluation of simulation abstractions for wireless sensor network mac protocols. *J. Wirel. Commun. Netw.*, 2010:24:1–24:2, Apr. 2010.

[2] H. N. Pham, D. Pediaditakis, and A. Boulis. From simulation to real deployments in wsn and back. In *IEEE WoWMoM 2007*, page 1–6, Jun. 2007.

[3] F. Österlind, A. Dunkels, T. Voigt, N. Tsiftes, J. Eriksson, and N. Finne. Sensornet checkpointing: Enabling repeatability in testbeds and realism in simulators. In *EWSN 2009*, Cork, Ireland, Feb. 2009.