Demo Abstract: EmuLink - Heterogeneous Sensor Network Simulation in Cooja

Joakim Eriksson[1], Niclas Finne[1], Nicolas Tsiftes[1], Thiemo Voigt[1], Mikael Gielda[2], and Peter Gielda[2]

[1] Swedish Institute of Computer Science, Sweden
{joakime,nfi,nvt,thiemo}@sics.se
[2] Ant Micro, Poland
{mgielda,pgielda}@antmicro.se

Abstract. Until now sensor network simulators have been limited when it comes to the variety of the hardware platforms that can be simulated. To enable simulation of heterogeneous sensor networks in Cooja, we present EmuLink—a component that connects different emulators into Cooja. In this demo, we demonstrate EmuLink's capability to simulate sensor networks combining nodes based on the TI MSP430 architecture and nodes based on the ARM Cortex-M3 architecture.

1 Introduction

With the rapid increase in the amount of wireless sensor nodes and other wireless devices forming heterogeneous networks, it becomes unfeasible to test real setups using physical hardware. While one can test systems and protocols on an abstract level by simulating wireless phenomena, such simulation alone is insufficient because software can be prone to bugs and unexpected interrelations. Simulating complicated wireless setups using exactly the same firmware image that will later be used on real wireless nodes is therefore crucial.

The Cooja wireless simulator [6] is widely used, but the range of hardware it can emulate has been limited. The original approach, in which MSPSim [2] was tightly integrated into the Cooja source code, limited the ability of running unmodified firmware images to nodes based on either 16-bit TI MSP430 CPUs using MSPSim or 8-bit Atmel AVR CPUs using Avrora.

Taking into consideration the growing popularity of low-power 32-bit CPUs, which are gaining a widespread adoption in the wireless sensor field [5], we need a way to integrate Cooja with an emulation framework that can support Cortex-M3. This was the reason for creating EmuLink, an abstract solution for interconnecting Cooja and hardware emulation software.

Although Cooja has proven to be useful for interoperability tests of heterogeneous software stacks before creating EmuLink [3, 4], the emulated hardware was limited to a single platform. By using EmuLink it becomes possible to take interoperability testing one step further and have both heterogeneous software and hardware.

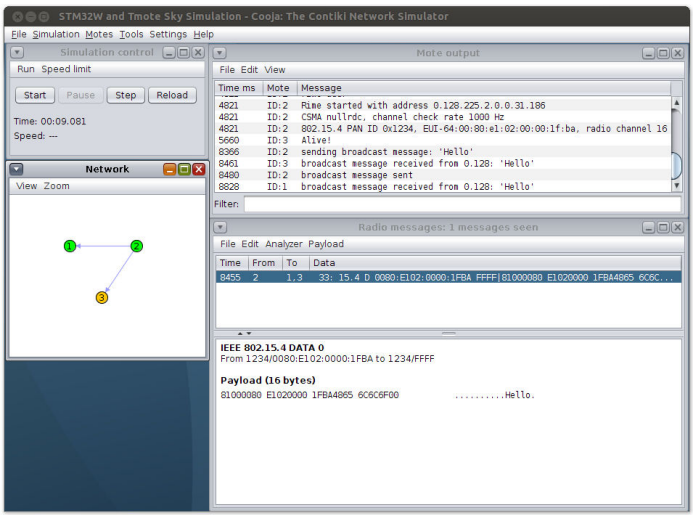

Fig. 1. Screenshot of Cooja running a simulation with two STM32W motes emulated using Prisma Emulator and one Tmote Sky mote emulated using MSPSim.

This demonstration exhibits the interconnection between MSPSim and Prisma Emulator through EmuLink. We show a simulated network of both types of nodes in Cooja, and show how we are able to send radio packets between the nodes using Contiki's network stack. The graphical user interface of Cooja, shown partly in Figure 1, allows direct interaction with the emulated nodes over the serial port, showing their actual execution as it would appear running on real hardware.

2 Architecture

Figure 2 shows our simulation architecture consisting of the Cooja sensor network simulator, MSPSim, Prisma Emulator, and EmuLink. In the following, we describe each individual component briefly.

Cooja/MSPSim

Cooja is a cross-level network simulator implemented in Java. Cooja handles the simulation of radio mediums and communication between nodes. The simulated nodes are implemented using a plugin architecture enabling various types of nodes. Node types range from abstract nodes implemented in Java to nodes executing firmware images using cycle-accurate hardware emulators.

MSPSim is a Java-based emulator of the TI MSP430 microprocessor series. MSPSim is tightly integrated in Cooja and provides cycle-accurate emulation, which enables development of timing-sensitive applications in Cooja. Exactly the same firmware that runs on a real sensor node can be loaded and executed in

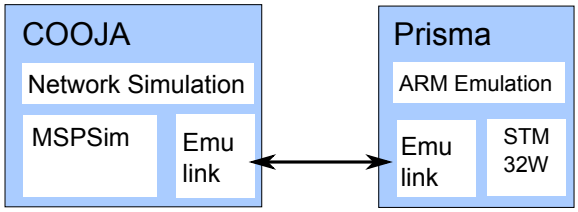

Fig. 2. EmuLink connects Cooja with external emulators such as the Prisma Emulator.

MSPSim, thereby providing a highly valuable development and debugging capability as well. Furthermore, we have earlier shown that Cooja/MSPSim enables accurate network-scale power profiling of sensor networks [2].

Prisma Emulator

The Prisma Emulator is partly based on QEMU and supports emulation of a wide variety of ARM microprocessors. The Prisma Emulator is extensible through Python plug-ins that make it possible to easily set up emulation of specific boards and System on Chip. For the Cooja integration we implement emulation of the STM32W platform, which is a Cortex-M3 with an integrated IEEE 802.15.4 radio.

The Prisma Emulator is implemented partly in native code, partly for the .NET framework. It executes in the Mono platform, which provides a cross-platform .NET development framework [1]. The different run-time environment used by Prisma Emulator makes it difficult to integrate into the Java-based Cooja platform. Hence, this initial hurdle in creating heterogeneous simulation support for Cooja is what motivated us to create the EmuLink component to connect emulators implemented in any language.

EmuLink

EmuLink consists of an EmuLink component in Cooja that connects external emulators into Cooja and EmuLink-enabled emulators that act as emulation servers. When running a simulation, Cooja will schedule the emulators for execution and deliver radio messages as well as serial data to and from the nodes that execute in the external emulators.

By using the concept of abstract emulation servers, EmuLink enables the use of emulators written in any programming language with Cooja and also makes it possible to distribute the hardware emulators on multiple computers to be able to scale the simulations to large networks.

3 Demo Setup

The demo consists of two laptops, one running Cooja, and the other running Prisma Emulator. We demonstrate a network simulation comprising internal

nodes emulated by MSPSim and external nodes emulated by Prisma Emulator. The simulation shows that Cooja with EmuLink handles both types of nodes and enables communication between them.

4 Conclusions and Future Work

EmuLink enables heterogeneous simulations in Cooja. This is shown using the Prisma Emulator and connecting it to Cooja with EmuLink. With EmuLink Cooja becomes a platform for interoperability testing on multiple hardware and software platforms. With the availability of Prisma Emulator, Cooja also becomes a simulation framework for the 32-bit Cortex-M3.

A limitation with the current implementation is that Prisma Emulator is driven by the system clock and runs in real time, thereby requiring the simulations to also run in real time. Another effect of this design is that the simulations become non-deterministic. We plan to improve this by driving the EmuLink-connected emulators with Cooja's simulation clock to enable the simulations to run at arbitrary speeds.

Acknowledgments

This work was partly funded by the Uppsala VINN Excellence Center for Wireless Sensor Networks WISENET and by SSF, the Swedish Foundation for Strategic Research.

References

1. Mono, the open source development platform based on the .NET framework. http://mono-project.com/. Visited 2012-12-15.
2. J. Eriksson, F. Österlind, N. Finne, A. Dunkels, N. Tsiftes, and T. Voigt. Accurate network-scale power profiling for sensor network simulators. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, Cork, Ireland, February 2009.
3. J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. José Marrón. Cooja/MSPSim: Interoperability testing for wireless sensor networks. In *SIMUTools 2009*, Rome, Italy, March 2009.
4. JeongGil Ko, Joakim Eriksson, Nicolas Tsiftes, Stephen Dawson-Haggerty, Jean-Philippe Vasseur, Mathilde Durvy, Andreas Terzis, Adam Dunkels, and David Culler. Beyond interoperability: Pushing the performance of sensor network IP stacks. In *SenSys '11*, November 2011.
5. JeongGil Ko, Kevin Klues, Christian Richter, Wanja Hofer, Branislav Kusy, Michael Bruenig, Thomas Schmid, Qiang Wang, Prabal Dutta, and Andreas Terzis. Low power or high performance? a tradeoff whose time has come (and nearly gone). In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, February 2012.
6. F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with Cooja. In *International Workshop on Practical Issues in Building Sensor Network Applications*, Tampa, Florida, USA, November 2006.