View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Swedish Institute of Computer Science Publications Database

Demo Abstract: Cooja TimeLine: A Power Visualizer for Sensor Network Simulation

Fredrik Österlind, Joakim Eriksson, Adam Dunkels
fros@sics.se, joakime@sics.se, adam@sics.se
Swedish Institute of Computer Science

Abstract

Power consumption is one of the most important factors in wireless sensor network research, but most simulators do not provide support for visualizing the power consumption of an entire sensor network. This makes it hard to develop, debug, and understand mechanisms and protocols based on power-saving mechanisms. We present Cooja TimeLine, an extension to Contiki's Cooja network simulator, that visualizes radio traffic and radio usage of sensor networks. Cooja TimeLine makes is possible to visually see the behavior of low-power protocols and mechanisms thereby increasing the understanding of the behavior of sensor networks. We see this as an important tool for the field moving forward.

1 Introduction

Power consumption is one of the most important factors in wireless sensor network research, but power consumption is not tangible and therefor difficult to understand. Simulators are a widely used tool to study, develop, and debug sensor networks. Yet sensor network simulators do not provide support for visualizing the power consumption of the networks that are simulated.

We present Cooja TimeLine, a module for the Contiki Cooja network simulator that visualizes both the power consumption and the network traffic of sensor networks. We have used Cooja TimeLine in the development of many of the power-saving mechanisms and protocols in the Contiki operating system.

In sensor networks, duty cycling MAC protocols are used to maintain a low power consumption of nodes. Even if the underlying principles behind power-saving MAC protocols are becoming widely understood [4], the implications of these protocols are not. For example, when a modern duty cycling protocol is used, nodes cannot overhear the transmissions of other nodes. Yet, many recent protocols are designed under the assumption that nodes can overhear each other. We argue that visualization tools like Cooja TimeLine greatly simplifies understanding and debugging of duty-cycled sensor network applications.

Copyright is held by the author/owner(s).
SenSys'10, November 3–5, 2010, Zurich, Switzerland.
ACM 978-1-4503-0344-6

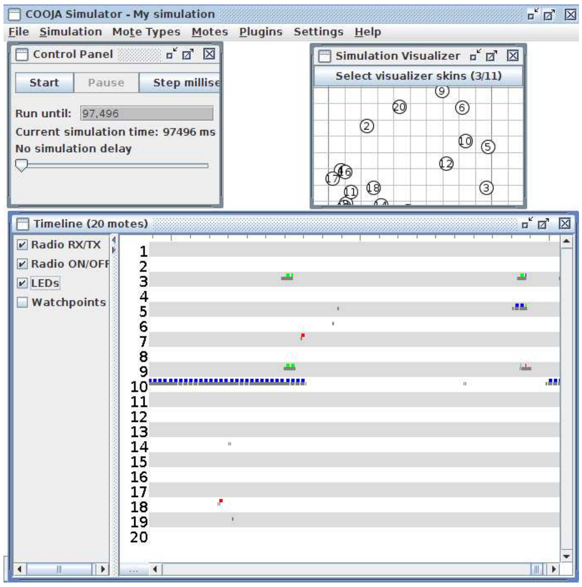

Figure 1. The Cooja TimeLine module in Cooja.

2 Cooja TimeLine

Cooja TimeLine shows a time line for each node in the simulation. On the time line, the power state of the radio transceiver of each node is shown in a color code: white if the transceiver is off, gray if it is on. Radio transmissions and receptions are shown in the same time line: transmissions are blue and receptions are green. Two simultaneous transmissions to a node, which results in radio interference, is shown in red. Transmissions and receptions are shown with bit granularity. See Figure 1 for a Cooja screenshot with a TimeLine module.

Cooja TimeLine observes the state of the emulation of the radio transceiver of all nodes in a simulation. As the simulation progresses, the contents of the window is scrolled to the left as simulated time passes.

The TimeLine module does not care about the specific type of radio that is emulated in each node. The architecture is general enough to support any type of emulated radio. We currently have implementations of one CC2420 radio emulator as well as an abstract radio that is not tied to any particular hardware platform.

3 Usage

We have been using the Cooja TimeLine module for several tasks in and around the Contiki operating system. We have found TimeLine to be useful for development and debugging of both low-level mechanisms [1, 2] and higher level protocols [3, 5]. We have identified three activities where TimeLine has been particularly useful to us: synchronization debugging, power debugging, and congestion debugging.

3.1 Visualizing Synchronization

Depending on the protocol under study, synchronization is either wanted or unwanted. Protocols that use synchronization will want their nodes to be synchronized. Other protocols run into unwanted or pathological synchronization conditions.

With the Cooja TimeLine module, synchronization is easy to visually identify. When nodes synchronize their transmissions or reception windows, the color markings will appear directly above and beneath each other in the TimeLine, making it easy to see. If synchronization is unwanted, it is normally easy to backtrack the simulation logs to see what caused the synchronization to occur. Likewise, if synchronization is wanted, but does not occur, it is possible to use the simulation logs or the debugger to investigate the causes. An explicitly synchronized network is shown in Figure 2.

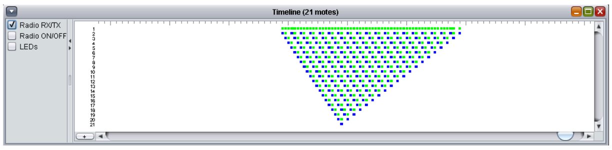

Figure 2. An explicitly synchronized multi-hop network.

3.2 Visualizing Power

Power is typically the main metric in sensor network research and power consumption is often dominated by the power consumption of the radio transceiver. As Cooja TimeLine tracks the state of each radio transceiver, it is possible to visually approximate the power consumption of an entire sensor network. This is shown in Figure 3, which shows a broadcast transmission with the low-power listening ContikiMAC radio duty cycling protocol.

If there is a problem with the power consumption in a network, e.g. because of a misbehaving protocol or application, it is possible to immediately visually identify the problem.

We have used Cooja TimeLine to debug and develop the duty cycling protocols in the Contiki operating system, such as ContikiMAC, Contiki X-MAC, and Contiki LPP. The original versions of Contiki X-MAC and Contiki LPP were developed without the support of Cooja TimeLine. When we first ran Cooja TimeLine on them, we witnessed several corner cases where the code failed to correctly turn off the radio. Cooja TimeLine significantly improved the implementations of these protocols.

3.3 Visualizing Congestion

Network congestion occurs when many transmitters want to send, despite there not being enough bandwidth available. In a low-power duty cycled network, congestion is made

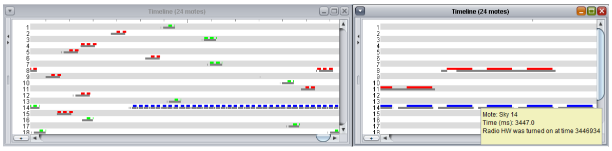

Figure 3. The radio duty cycle is visualized.

worse by the fact that a single transmission can occupy a significant amount of bandwidth. Specifically, a single-hop broadcast with an asynchronous duty cycling protocol results in a large number of link-layer transmissions. When many simultaneous transmitters send broadcasts, congestion occurs.

Congestion is next to impossible to debug in a testbed or a real deployed network, because radio transmissions cannot be seen. With Cooja TimeLine, however, congestion conditions are easily seen. By using the zoom functionality of the TimeLine module, it is possible to visually inspect the traffic intensity over both short and long timescales. Congestion is thus easily identified, as shown in Figure 4.

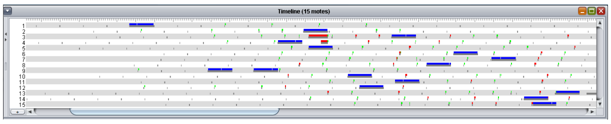

Figure 4. A congested network.

4 Conclusions

We present Cooja TimeLine, a graphical module for the Contiki Cooja network simulator. Cooja TimeLine draws a moving graph of the state of on-board peripherals of each node as well as the radio traffic being sent between nodes. This makes it easy to visually inspect network traffic, power consumption, and synchronization in a simulated sensor network. We have been using Cooja TimeLine for both low-level and high-level mechanisms and protocols in the Contiki operating system. We have found CoojaTimeLine to be particularly useful for debugging synchronization, power consumption, and congestion, but there are many other uses as well. We see Cooja TimeLine as an important tool for developing an understanding of the behavior of low-power duty cycled sensor networks, which is not possible using traditional debuggers or simulators.

5 References

[1] F. Österlind and A. Dunkels. Approaching the maximum 802.15.4 multi-hop throughput. In *Proceedings of the Workshop on Hot Topics in Embedded Networked Sensor Systems (HotEmnets)*, June 2008.

[2] F. Österlind, N. Wirström, N. Tsiftes, N. Finne, T. Voigt, and A. Dunkels. StrawMAN: Making Sudden Traffic Surges Graceful in Low-Power Wireless Networks. In *Proceedings of the 2010 ACM HotEMNETS Workshop on Hot Topics in Embedded Networked Sensosr*, Killarney, Ireland, June 2010.

[3] N. Tsiftes, J. Eriksson, and A. Dunkels. Low-Power Wireless IPv6 Routing with ContikiRPL. In *Proceedings of the International Conference on Information Processing in Sensor Networks (ACM/IEEE IPSN)*, Stockholm, Sweden, April 2010.

[4] M. Welsh. The next decade of sensor networking. Keynote at the 2010 European Workshop on Sensor Networks (EWSN).

[5] D. Yazar and A. Dunkels. Efficient Application Integration in IP-Based Sensor Networks. In *Proceedings of the ACM BuildSys 2009 workshop, in conjuction with ACM SenSys 2009*, November 2009.