Hardware and software system-level simulator for wireless sensor networks

David Navarro, Wan Du, Fabien Mieyeville, Laurent Carrel

Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 5 (2010) 228–231
Procedia Engineering
www.elsevier.com/locate/procedia

Proc. Eurosensors XXIV, September 5-8, 2010, Linz, Austria

Hardware and Software System-Level Simulator for Wireless Sensor Networks

D. Navarro[a], W. Du[a], F. Mieyeville[a], L. Carrel[a]

[a] *Université de Lyon, Institut des Nanotechnologies de Lyon (INL), UMR5270 - CNRS, Ecole Centrale de Lyon, Ecully, F-69134, France*

Abstract

Wireless Sensor Network is a large-scale network of resource-constrained (energy, memory and processing) sensor nodes that are deployed at different locations, in order to monitor physical or environmental conditions. A design problem is that a lot of sensor node platforms exist, and many MAC-layer and protocol configurations can be used. It is also difficult to explore design space to check for feasibility at early design stage.
Focusing on widely used IEEE 802.15.4 standard, our simulator - called IDEA1- enables designers to check, through an easy to use graphical environment, the possible use of actual and future systems. That SystemC simulator is also composed of a library (that is periodically improved) of electrical devices and software stacks. It permits to find which hardware system to choose and which software to develop to best fit application requirements.

Keywords: wireless sensor netwok; WSN; systemC; simulator; hardware; software; 802.15.4

1. Wireless sensor networks architecture and simulators

Many applications, such as environmental data collection, security monitoring, logistics or health, use communicating and distributed sensory systems [1]. These radiofrequency-based communicating systems are called Wireless Sensor Networks (WSN). WSN are large-scale networks of resource-constrained sensor nodes (electronic systems). Limited resources are of different kinds: energy, memory and processing. The sensor nodes cooperatively monitor and transmit physical or environmental conditions, such as temperature, vibration, pressure etc. They are typically composed of one or more sensors, an 8-bit or 16-bit microcontroller, a non-volatile memory, a radiofrequency transceiver and a battery. Fig. 1 shows typical sensor node architecture.

Brand examples of often used devices are: ATMEL, Texas Instruments or Microchip microcontrollers and Texas Instruments, ATMEL, Freescale, or ST-Microelectronics radiofrequency transceivers. Linux systems composed of 32-bit RISC processors exist – like the well known Crossbow's Stargarte platform - but energy consumption is prohibitive and autonomy is largely affected, thus relegating these products to the border of the WSN field. We do not consider such systems, and we do focus on long-autonomy systems.

doi:10.1016/j.proeng.2010.09.089

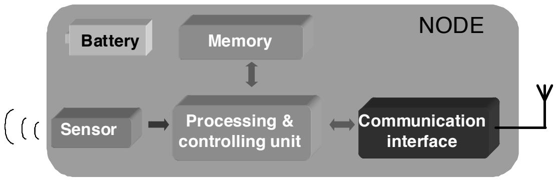


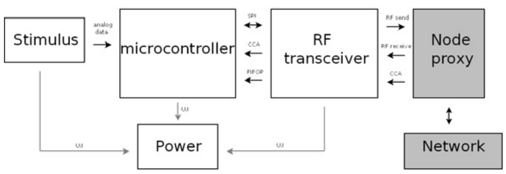
Fig. 1. Wireless sensor node architecture

Wireless Sensor Networks interconnect (topologies and network hierarchy) is indeed inspired from wireless telecommunication networks, and it can be complex. We consider the often used IEEE 802.15.4 standard [2] that is widespread, in many WSN commercial or custom platforms. Although complex topologies exist, such networks are dedicated to low power and low data rate applications, mainly for physical and environmental remote measurements. Wireless Sensor Networks design is a difficult task, because designer has to develop a network at system level, with low level (at sensor node) hardware and software constraints. CAD tools would also be required to make system-level (hardware and software) simulations, taking low-level parameters into account. In this paper, we propose a graphical SystemC-based Wireless Sensor Network simulator that supports several hardware devices and considers the IEEE 802.15.4 standard.

Many simulators have been developed last few years [3-8], but most of them are restricted to specific hardware or precisely focus on either network level or node level. They can be broadly divided in two categories: network simulators enhanced with node models (NS-2 [6] and OMNeT++ [7]), and node simulators enhanced with network models (Avrora [8], or SCNSL [9]). In first category, simulators are not sensor platform specific and they are too high level for hardware considerations. In second category, simulators are better suited for embedded system designers, requiring precise low level models for top-down (network to node) approach, but they suffer of too low-level aspects.

2. Proposed system-level simulator

We inspired our work from the SCNSL library [9], a networked embedded systems simulator, written in SystemC and C++. SystemC is widely used in electronics community, and it is able to model hardware, software, and network. SCNSL has been deeply modified to precisely model hardware and to implement the whole IEEE 802.15.4 standard with many configurations. In original SCNSL, three modules exist: node (in SystemC), node-proxy (in SystemC) and network (in C++). The node-proxy can access the network; it is the interface between nodes and the network. In many simulators -as in original library- , hardware is modeled at node level. As shown in Fig. 2.(a), we modeled a more accurate physical layer.



Microcontroller list	Radiofrequency Transceiver list
ATMEL ATMega128 Microchip 16LF88	Texas Instrument CC2420 Texas Instrument CC1000 Microchip MRJ24J40

Fig. 2. (a) Wireless sensor node architecture (b) modeled hardware devices

Indeed, hardware devices are more detailed and electrical parameters are taken into account. At the whole, several microcontrollers and several radiofrequency transceivers can be selected from our simulator library (Fig. 2.(b)).

The stimulus block generates analog sensor data towards the microcontroller. Microcontroller and radiofrequency transceiver are modeled separately, so that designers can switch devices. These two parts communicate through SPI (Serial Peripheral Interface) interfaces. Microcontroller is the central unit for processing and controlling purposes. In our typical case, microcontroller initializes the radiofrequency transceiver, then it reads (converts) data from sensor, and communicates data to radiofrequency transceiver. Radiofrequency transceiver model contains different working states (receive, transmit, idle, sleep), and several operating modes. In our model, the microcontroller can configure some parameters of physical (PHY) and MAC layers in the radiofrequency transceiver registers (IEEE 802.15.4 - compliant). Both IEEE 802.15.4 non-beacon and beacon modes have been implemented in our model, as shown in Fig. 3.

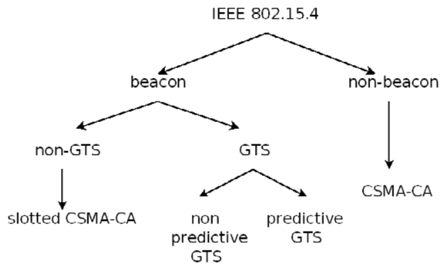


Fig. 3. Modeled IEEE 802.15.4 operating modes

Non-beacon mode is the simplest mode to study and to model. It is based on a channel free access and packet-based philosophy. When a node has to send data, it senses channel, then sends data if channel is free. If channel is in use, it waits a random time (called back-off time) and then checks for free channel again. That method is CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance). Beacon mode is a synchronized mode: the network coordinator (network head) sends synchronization packets to inform nodes when they can communicate. That mode is also channel-based, inspired from the well known TDMA (Time Division Multiple Access) method. Time is organized according to a superframe that is defined by the network coordinator. Several beacon-mode methods exists: slotted CSMA-CA, non-predictive GTS and predictive GTS [2]. Slotted CSMA-CA is a CSMA-CA based communication, within a given slot time. In non-predictive GTS, nodes that want to communicate send a GTS request to the coordinator during a first time slot (the Contention Access Period). Then, nodes are allowed to communicate during a following time slot (the Contention Free Period). In predictive GTS, the list and the communication order of nodes are fixed, at least for one superframe.

A power module has been implemented. It monitors electrical power and energy consumed by sensor, microcontroller and radiofrequency transceiver. Different energy-saving (sleep) modes, data flow and global behavior can also be co-designed according to power constraints.

The user interface is shown in Fig. 4.(a) below. It is composed of different sub-windows. The information appears graphically in the right window, to clearly see the network topology. Each node (that appears as a black dot in the interface) and coordinator (red dot) is characterized with a spatial position. Lines between nodes represent possible communications according to position, transmit power and receive sensitivity. When parameters are changed, the graphical viewer refreshes the possible communications (lines). For this early version, free space communications are considered. Focus is set on communication capabilities and data rate, not on mechanical or electromagnetic environments. Hardware parameters are some of selectable microcontrollers and radiofrequency devices. At higher level, one of the IEEE 802.15.4 configurations and superframe size can be selected. Nodes sensors sampling rate and payload of packets can also be changed.

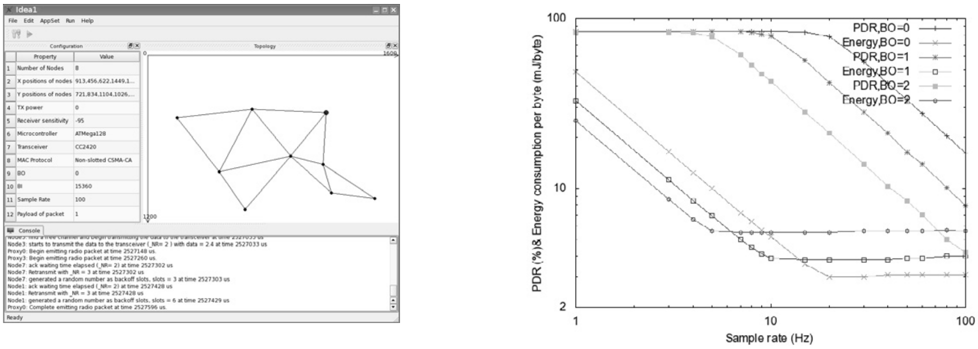


Fig. 4.(a) Simulator graphical interface (b) Output results

By clicking on the launch button in graphical interface, a SystemC simulation is launched in background. Simulation log is displayed in the bottom window of graphical interface, and a timing trace (VCD) viewer is opened. Output log files are also generated. From these results, we can explore design space for best solution.

As test example, we simulated an 8 nodes network. We chose ATMEL ATMega128 and Texas Instruments CC2420 as hardware. As IEEE 802.15.4 data-rate is low (250 Kb/s), a systematic trade-off between payload (number of sent data bytes per packet), sampling rate (of Analog to Digital Converter) and packet delivery rate has to be explored. Packet Delivery Rate (PDR) is the ratio of number of successful packets over the total number of sent packets. From Fig. 4.(b), we can identify for these (changeable) devices which IEEE 802.15.4 is most interesting depending on Packet Delivery Rate (PDR), energy and sampling rate.

3. Conclusion

A behavioral system-level simulator for Wireless Sensor Networks design space exploration has been presented. Its graphical interface permits to easily simulate and compare several IEEE 802.15.4 configurations on many interchangeable (and parameterized) hardware devices. System-level communications can be validated (with the help from text or curves outputs), according to specific hardware devices characteristics. The authors are now working on a specific test-case that is being simulated and programmed.

References

[1] M. Horton, J. Suh, A vision for wireless sensor networks, IEEE Microwave Symposium Digest, 2005.

[2] IEEE 802.15 WPAN Task Grup 4, IEEE 802.15 Part 15.4-2006 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), http://www.ieee802.org/15/pub/TG4.html

[3] S. Park, A. Savvides, M. B. Srivastava, SensorSim: A Simulation Framework for Sensor Networks, Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2000.

[4] P. Levis, N. Lee, M. Welsh,D. Culler, Tossim: Accurate and scalable simulation of entire tinyos applications, Embedded Networked Sensor Systems, 2003.

[5] M. Varshney, D. Xu, M. Srivastava, R. Bagrodia, sQualNet: An Accurate and Scalable Evaluation Framework for Sensor Networks, Information Processing in Sensor Networks, 2007.

[6] S. McCanne and S. Floyd, Network Simulator NS-2, http://www.isi.edu/nsnam/ns, 2010.

[7] A. Varga, The OMNeT++ discrete event simulation system, European Simulation Multiconference, 2001.

[8] B. Titzer, D. Lee, and J. Palsberg, Avrora: Scalable sensor network simulation with precise timing, Information Processing in Sensor Networks, 2005.

[9] F. Fummi, D. Quaglia, F. Stefanni, "A SystemC-based Framework for Modeling and Simulation of Networked Embedded Systems", Forum on Specification and Design Languages, 2008.