Heterogeneous Wireless Sensor Network Simulation

D. Navarro, M. Galos, F. Mieyeville, W. Du
Université de Lyon, Institut des Nanotechnologies de Lyon (INL)
UMR5270 - CNRS, Ecole Centrale de Lyon, Ecully, F-69134, France
David.Navarro@ec-lyon.fr

Abstract - **Based on our previous work on the development of a Wireless Sensor Network (WSN) simulation platform, we present here its ability to run simulations on heterogeneous nodes. This platform allows system-level simulations with low level accurate models, with graphical inputs and outputs to easily simulate such distributed systems. In the testbed we consider, the well known IEEE 802.15.4 standard is used, and different microcontrollers units (MCU) and radiofrequency transceivers compose the heterogeneous nodes. It is also possible to simulate complex networks or inter-acting networks; that is a more realistic case, as more and more hardware devices exist and standards permit their interoperability. This simulation platform can be used to explore design space in order to find the hardware devices and IEEE 802.15.4 algorithm that best fit a given application. Packet Delivery Rate (PDR) and packet latency can be evaluated, as other network simulators do. Energy consumption of sensor nodes is detailed with a very fine granularity: partitioning over and into hardware devices that compose the node is studied.**

Keywords – Wireless Sensor Network, WSN, heterogeneous, simulation, model, SystemC

I. INTRODUCTION

Wireless Sensor Networks (WSN) are widespread sensory systems. They are used in a variety of applications, such as environmental data collection, security monitoring, logistics or health [1]. Wireless Sensor Networks are large-scale networks of resource-constrained sensor nodes that are deployed at different locations. Limited resources are: energy, memory and processing. The sensor nodes cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration or pressure. Because of autonomy requirements, they have a specific architecture; they are typically composed of one or more sensors, an 8-bit or 16-bit MCU, sometimes a non-volatile memory, a radiofrequency transceiver and an energy supply. Typical hardware structures are detailed in Figure 1, where we can recognize two heterogeneous nodes. Manufacturers of WSN hardware include: ATMEL, Texas Instruments or Microchip MCU 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 Stargate platform - but energy consumption is prohibitive and autonomy is largely affected, thus relegating these products to the border of the WSN field, often for high data rate applications. We do not consider such systems, and we focus on several months of battery life systems.

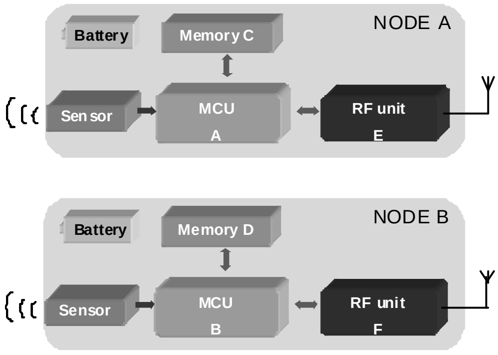


Figure 1. Typical node architectures in a Wireless Sensor Network (heterogeneous network)

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 are also required to make system-level (hardware and software) simulations, taking low-level parameters into account. It is what our simulator –IDEA1- permits. Moreover, we detail in this paper a new feature of our simulation platform, which supports heterogeneous sensor nodes.

II. WIRELESS SENSOR NETWORKS SIMULATORS

Many simulators have been developed last few years [2-6]. Unfortunately, most of them are restricted to specific hardware or precisely focus on either network level or node level. Research on sensor network evaluation can be broadly divided in two categories: network simulators enhanced with node models, and node simulators enhanced with network models. A thorough exploration of this field is given in [7].

Typical network simulators are general purpose network simulator, such as Network Simulator NS-2 [5] and OMNeT++ [6] (and their declinations). The problem is these interesting network simulators are not sensor platform specific and they are also too high level for hardware considerations. Moreover, there is no separation between computation and communication models. That modeling is also not suitable for hardware replacements and explorations. Then, such simulators do not have accurate energy models [8].

Node simulators refer to precise hardware description, with a synchronization strategy among the nodes, such as Avrora [9], or SCNSL - SystemC Network Simulation Library [10]. These simulators are better suited for embedded system designers, requiring precise low level models for top-down (network to node) approach. SCNSL is a networked systems simulator, written in SystemC and C++. Because SystemC is a C++ class library, it has the advantage to be able to model hardware, software, and network. SystemC is a classical and widely used modeling language in electronic systems design and particularly in System-On-Chip design where it enables hardware/software communication and protocol exploration. Our simulation platform is also based on SystemC and C++, and SCNSL was the starting point of our work.

III. PROPOSED SYSTEM-LEVEL MODELS

Our models architecture is close to hardware architecture, as Figure 2 (compared to Figure 1) shows. Software and the whole IEEE 802.15.4 standard with many configurations are modeled too.

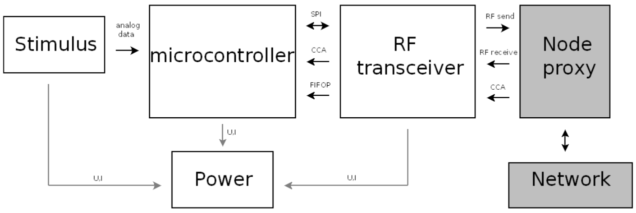


Figure 2. Node model architecture

The stimulus block generates analog sensor data towards the MCU. MCU and radiofrequency transceiver are modeled separately, so that designers can switch these interchangeable devices. These two parts communicate through SPI (Serial Peripheral Interface) interfaces.

MCU is the central unit for processing and controlling purposes. In our typical case, MCU initializes the radiofrequency transceiver, then it reads (converts) data from sensor, and communicates data to radiofrequency transceiver. Switch between such architectures is done by changing some parameters. MCU model can for example switch from ATMEL ATMega128 to Microchip PIC16LF88. Figure 3 shows the generic FSM (Finite State Machine) we have implemented for MCU. Parameters depend on the MCU itself and from the radiofrequency transceiver (according for example to its hardware support of IEEE 802.15.4 or no).

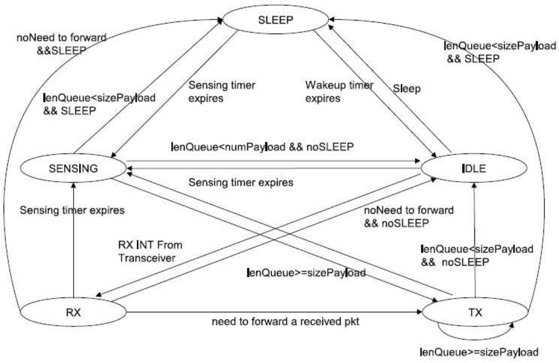

SLEEP
noNeed to forward &&SLEEP
lenQueue<sizePayload && SLEEP
Sensing timer expires
Wakeup timer expires
Sleep
lenQueue<sizePayload && SLEEP
SENSING
lenQueue<numPayload && noSLEEP
IDLE
Sensing timer expires
Sensing timer expires
noNeed to forward && noSLEEP
RX INT From Transceiver
lenQueue<sizePayload && noSLEEP
lenQueue>=sizePayload
RX
need to forward a received pkt
TX
lenQueue>=sizePayload

Figure 3. Generic FSM for MCU

Radiofrequency transceivers are modeled individually because of their complexity and wide differences (that would make difficult a generic FSM). Below are drawn two FSM examples of two well known radiofrequency transceivers.

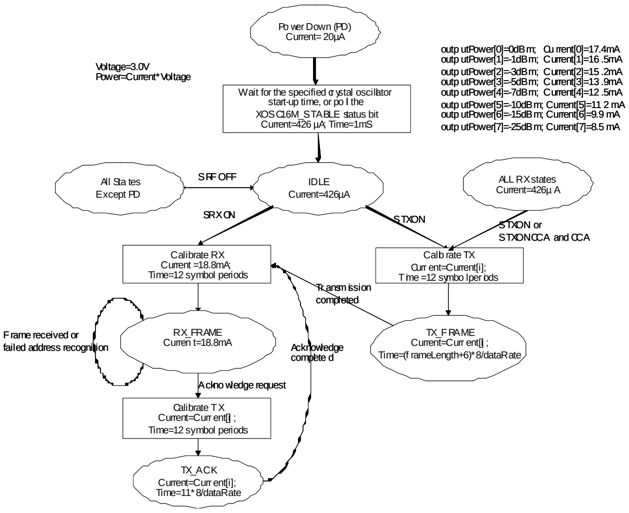

Power Down (PD)
Current= 20µA
Voltage=3.0V
Power=Current* Voltage
outputPower[0]=0dBm; Current[0]=17.4mA
outputPower[1]=-1dBm; Current[1]=16.5mA
outputPower[2]=-3dBm; Current[2]=15.2mA
outputPower[3]=-5dBm; Current[3]=13.9mA
outputPower[4]=-7dBm; Current[4]=12.5mA
outputPower[5]=-10dBm; Current[5]=11 2 mA
outputPower[6]=-15dBm; Current[6]=9.9 mA
outputPower[7]=-25dBm; Current[7]=8.5 mA
Wait for the specified crystal oscillator start-up time, or pol l the XOSC16M_STABLE status bit
Current=426 µA; Time=1mS
All States Except PD
SRFOFF
IDLE
Current=426µA
ALL RX states
Current=426µ A
SRXON
STXON
STXON or STXONCCA and CCA
Calibrate RX
Current =18.8mA;
Time=12 symbol periods
Calibrate TX
Current=Current[i];
Time =12 symbol periods
Transmission completed
Frame received or failed address recognition
RX_FRAME
Current=18.8mA
Acknowledge completed
TX_FRAME
Current=Current[i] ;
Time=(frameLength+6)*8/dataRate
Acknowledge request
Calibrate TX
Current=Current[i] ;
Time=12 symbol periods
TX_ACK
Current=Current[i];
Time=11*8/dataRate

Figure 4. TI CC2420 non slotted CSMA-CA Finite State Machine

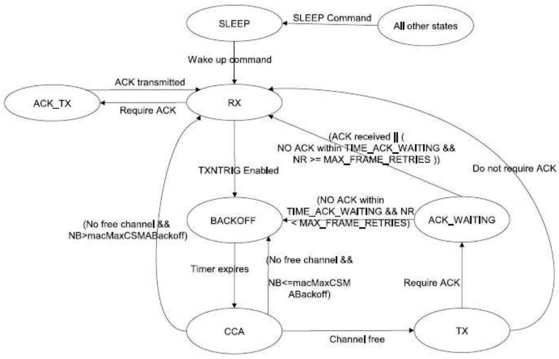

SLEEP
SLEEP Command
All other states
Wake up command
ACK transmitted
ACK_TX
Require ACK
RX
(ACK received || (NO ACK within TIME_ACK_WAITING && NR >= MAX_FRAME_RETRIES))
TXNTRIG Enabled
Do not require ACK
(NO ACK within TIME_ACK_WAITING && NR < MAX_FRAME_RETRIES)
(No free channel && NB>macMaxCSMABackoff)
BACKOFF
ACK_WAITING
Timer expires
(No free channel && NB<=macMaxCSMABackoff)
Require ACK
CCA
Channel free
TX

Figure 5. MRF24J40 non slotted CSMA-CA Finite State Machine

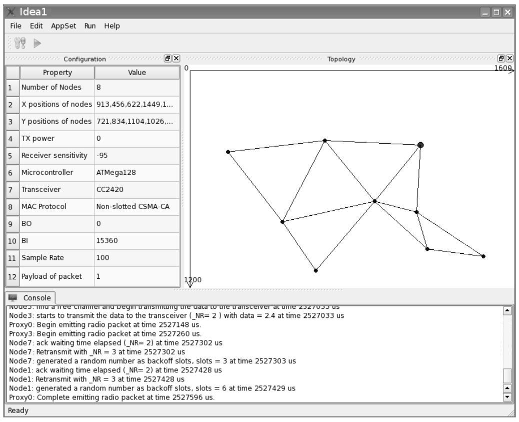
As a whole, several MCU and several radiofrequency transceivers can be selected; the current library is detailed in table I. Each MCU can be mapped to each radiofrequency transceiver. Each radiofrequency transceiver includes the whole IEEE 802.15.4 standard. As it has been previously published, all of these models have been validated with experimental measurements on many testbeds [11], and are 6% accurate.

TABLE I. MODELED DEVICES IN SIMULATOR LIBRARY

MCU	Radiofrequency transceivers
ATMEL ATMega128 Microchip 16LF88 Texas Instruments MSP 430	Texas Instruments CC2420 Texas Instruments CC1000 Microchip MRJ24J40

IV. SIMULATOR

The presented models can be used to simulate heterogeneous network communications at system level. To help non-specialists to use easily the simulation tool, we developed a graphical interface that is shown in the figure below.



	Property	Value
1	Number of Nodes	8
2	X positions of nodes	913,456,622,1449,1...
3	Y positions of nodes	721,834,1104,1026,...
4	TX power	0
5	Receiver sensitivity	-95
6	Microcontroller	ATMega128
7	Transceiver	CC2420
8	MAC Protocol	Non-slotted CSMA-CA
9	BO	0
10	BI	15360
11	Sample Rate	100
12	Payload of packet	1

```
Node3: find a free channel and begin transmitting the data to the transceiver at time 2527033 us
Node3: starts to transmit the data to the transceiver (_NR= 2 ) with data = 2.4 at time 2527033 us
Proxy0: Begin emitting radio packet at time 2527148 us.
Proxy3: Begin emitting radio packet at time 2527260 us.
Node7: ack waiting time elapsed (_NR= 2) at time 2527302 us
Node7: Retransmit with _NR = 3 at time 2527302 us
Node7: generated a random number as backoff slots, slots = 3 at time 2527303 us
Node1: ack waiting time elapsed (_NR= 2) at time 2527428 us
Node1: Retransmit with _NR = 3 at time 2527428 us
Node1: generated a random number as backoff slots, slots = 6 at time 2527429 us
Proxy0: Complete emitting radio packet at time 2527596 us.
```

Figure 6. Simulator graphical user interface

The user interface is composed of different sub-windows. A graphical viewer shows spatial position of nodes and the possible communications according to locations, power of transmission and sensitivity of receiver. Hardware parameters are some of selectable MCU and radiofrequency transceivers. One of the many IEEE 802.15.4 configurations and superframe parameters can be selected. Nodes sensors sampling rate and payload of packets can also be set. 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 (Value Change Dump format: VCD) viewer is opened. Output log files are also generated. Concrete examples are given in the following section. From these results, we can explore design space for best solution.

V. HETEROGENEOUS SIMULATION RESULTS

Heterogeneous support in simulators with fine and accurate hardware and software models is necessary but few simulators support this feature [12][13]. One reason is the need of a complex organization of models. As test example, we simulated a 9 nodes network: 1 coordinator and 8 nodes composed of Microchip PIC16LF88 and ATMEL AVR ATMega128L MCU and Microchip MRF24J40 and Texas Instruments CC2420 radiofrequency transceiver, as specified in table II.

TABLE II. NODES DEVICES FOR TESTBED

WSN device	MCU	RF Transceiver
Coordinator	ATMega128	CC2420
Nodes 0..3	PIC16LF88	MRF24J40
Nodes 4..7	ATMega128	CC2420

Nodes sense the environment periodically and transmit data over the network. Each transmission (packet) includes 2 data bytes (payload). Sensor nodes enter sleep mode as long as they can, coordinator doesn't enter sleep mode. IEEE 802.15.4 classical data-rate (theoretically 250 Kb/s) is used. A non beacon CSMA-CA with no acknowledge is used, but all of the IEEE 802.15.4 can be chosen for more complex applications and wider exploration. Previous works on non heterogeneous (homogeneous) WSN with this simulation platform have already been published [14] to validate accuracy of the models by experimental measurements.

By clicking on the launch button in graphical interface, SystemC batch command is launched in background, and simulation log is displayed in the bottom window. Then, a trace (VCD) viewer is opened. A part of VCD trace in shown in Figure 7.

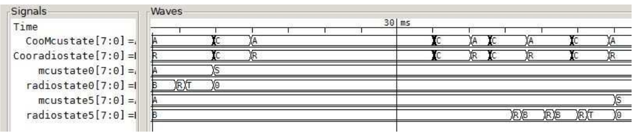


Figure 7. Extract of the output VCD file, focus on coordinator and nodes 0 and 5 (MCU and radiofrequency transceiver states)

We can observe the coordinator's and nodes' MCU and radiofrequency transceiver states. It is possible to monitor more signals in order to see the channel usage and data transfers from sensor to radiofrequency transceiver through MCU on each node, and data from radiofrequency transceiver to MCU on coordinator. It is then possible to monitor latency and packet delivery rate (PDR) from these curves. From log file, PDR and latency can be precisely displayed; energy can be read to draw graphs such as these in the following figures.

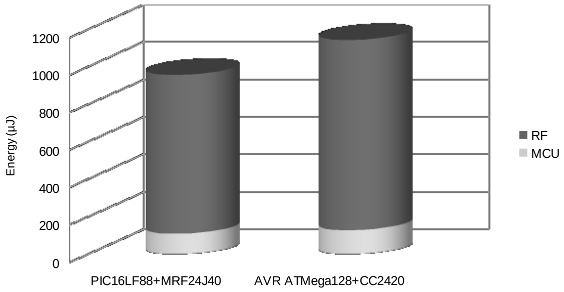


Figure 8. Heterogeneous nodes energy consumption

Figure 8 shows energy partitioning between MCU and radiofrequency transceiver for 2 heterogeneous nodes (node 0 and node 5). We can also see the (relative) low impact of MCU energy compared to energy consumed by radiofrequency transceiver.

It is moreover possible to have finer granularity and to detail energy consumption of each block within hardware devices. Figure 9 shows energy spent during SPI communications, active (CPU), and sleep states. It is to note that radiofrequency transceiver impacts active duration of MCU. Indeed, in that example, CC2420 transceiver just modulates the packet, MCU has all in charge. It has for example to check for free channel, to use delays for backoffs, to generate IEEE 802.15.4 compliant packets, to take acknowledge in charge if activated, etc. More SPI communications are also required. On the other hand, MRF24J40 is a more autonomous circuit, as it supports all the aspects above in hardware, MCU is also less active.

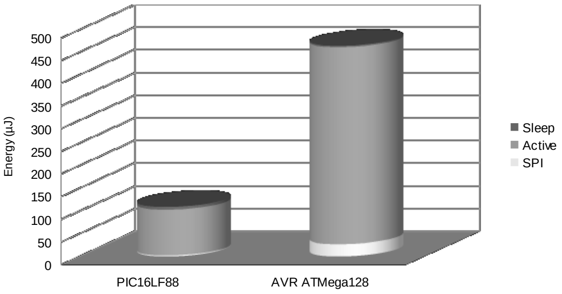


Figure 9. MCU energy consumption comparison

With the same fine granularity, it is possible to detail states of radiofrequency transceivers, as shown in Figure 10.

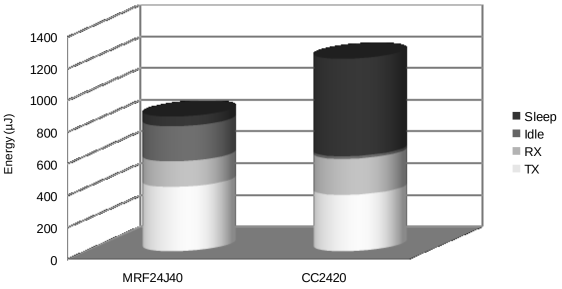


Figure 10. Radiofrequency transceiver energy consumption comparison

For each radiofrequency transceiver, this figure shows it is possible to monitor energy consumed during sleep mode, idle, receiving (RX) and transmitting (TX). According to the testbed (or real application) states durations, it is also possible to optimize total energy, with such a deep exploration.

VI. Conclusion

Heterogeneous support of our system-level simulator for Wireless Sensor Networks has been presented. This simulator is written in SystemC, which combines advantages of being a widely used language in electronic systems design flow, and permitting hardware and software modeling. The simulator graphical interface permits to easily configure a network and the sensor nodes characteristics, to obtain easy to read waveforms and easy to process output logs. As the whole IEEE 802.15.4 and many hardware devices are modeled, it is possible to simulate and compare IEEE 802.15.4 algorithms on many interchangeable (and parameterized) hardware devices (even by mixing them within a network) in order to run design space exploration. Classical network simulators outputs - Packet Delivery Rate (PDR) and packet latency - are accessible; as well as accurate timing, accurate and detailed power consumption of hardware devices.

References

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

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

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

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

[5] S. McCanne and S. Floyd, “Network Simulator NS-2”, http://ww.isi.edu/nsnam/ns, 2010.

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

[7] W. Du, D. Navarro and F. Gaffiot, “Towards a Taxonomy of Simulation Tools for Wireless Sensor Network”, International Conference on Simulation Tools and Techniques, 2010.

[8] F. Chen, I. Dietrich, R. German and F. Dressler, “An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++”, PIK Volume 32, Pages 133–138, 2009.

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

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

[11] F. Mieyeville, W. Du, I. Daikh and D. Navarro, "Wireless Sensor Networks for active control noise reduction in automotive domain", 14th International Symposium on Wireless Personal Multimedia Communications, 2011.

[12] F. Fummi, G. Perbellini, D. Quaglia, and A. Acquaviva, “Flexible energy-aware simulation of heterogeneous wireless sensor networks,” Conference on Design, Automation and Test in Europe, 2009.

[13] L. Girod, T. Stathopoulos, N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer, “A system for simulation, emulation, and deployment of heterogeneous sensor networks,” Int. Conf. on Embedded Networked Sensor Systems, 2004.

[14] F. Mieyeville, D. Navarro, W. Du and M. Galos, "Energy-centric simulation and design space exploration for Wireless Sensor Networks", CRC Press, Taylor & Francis Group, Book in Press, 2012.