2011 Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems

IDEA1: A Validated SystemC-Based Simulator for Wireless Sensor Networks

Wan Du, David Navarro, Fabien Mieyeville, and Ian O'connor
Heterogeneous Systems Design Group
Lyon Institute of Nanotechnology, University of Lyon, France
wan.du@ec-lyon.fr

***Abstract*—This paper presents IDEA1, a validated SystemC-based simulator for WSNs. It allows the system-level performance evaluation (e.g., packet transmission and energy consumption) with elaborate models of sensor nodes. IDEA1 uses a clock-based synchronization mechanism to support simulations with cycle accurate communication and approximate time computation. Its accuracy has been validated by a testbed of 9 nodes. The average deviation between the IDEA1 simulations and experimental measurements is 5.9%. The performances of IDEA1 have also been compared with NS-2, one of the most widely used simulators in WSN research. To provide a similar result (deviation less than 5%) at the same abstraction level, the simulation of IDEA1 is 2 times faster than NS-2. Moreover, with the hardware and software co-simulation feature, IDEA1 provides more detailed modeling of sensor nodes than NS-2.**

Keywords- wireless sensor networks; modeling and simulation; systemC; validation

I. INTRODUCTION

The constraints of small size and low cost of sensor nodes result in the limited energy supply. In order to extend the network lifetime, many efforts are taken to reduce the energy consumptions of Wireless Sensor Networks (WSN). Therefore, it is necessary to accurately predict the energy consumption while designing novel WSN applications and protocols. The accurate energy prediction requires detailed models of the hardware and software (HW/SW) of sensor nodes. Lots of simulation tools for WSN have been developed in recent years. They can be briefly divided into three categories: network simulators, node emulators and node simulators. A more detailed analysis of the existing WSN simulators can be found in our former work [1].

Many network simulators, such as NS-2 [2] and OMNeT++ [3], have been used in WSN simulations. A NS-2 IEEE 802.15.4 model is developed in [4], while an energy model is added in [5]. Generally, the network simulators have the advantages of extensibility, heterogeneity support and easy-to-use. However, the energy consumption estimation is usually based on some simple assumptions; for example, the processor and Radio-frequency (RF) transceiver have the same operating states.

Several Operating System (OS) emulators and Instruction Set Simulators (ISS) are developed to emulate the embedded software execution of sensor nodes. TOSSIM [6] is an OS emulator designed for the execution of TinyOS applications. It offers a controlled environment facilitating the development of algorithms and the study of system behaviors. Avrora [7] is an instruction-level cycle accurate emulator written in Java which uses an ISS of AVR processor as the simulation kernel. Node emulators provide a high timing accuracy of software execution. However, they are generally constrained to specific pre-defined hardware platforms or operating systems.

In order to provide accurate energy consumption estimation at system level, a novel SystemC-based WSN simulator named IDEA1 (hIerarchical DEsign plAtform for sensOr Networks Exploration) is developed. It supports the hardware and software (HW/SW) co-simulation of sensor nodes. It allows rapid performance evaluation of WSN applications and protocols at system-level. SystemC [8] is a C++ class library for system and hardware design. It is a System-Level Description Language (SLDL) which provides native supports to model concurrency, structural hierarchy, interrupts and synchronization of embedded systems [9]. Several SystemC-based WSN simulators have been developed recently, such as ATLeS-SN [10] and SNOPS [11]; however, to the best of our knowledge, IDEA1 is the first one that has been validated with experimental measurements and evaluated comprehensively by comparing with other simulators.

A testbed of 9 sensor nodes has been built to validate the accuracy of IDEA1. The deviation of IDEA1 simulations and the experimental measurements is acceptable by the general system-level simulations. The performances of IDEA1 have also been compared with NS-2 which is the most used simulator in Mobile Ad hoc NETwork (MANET) research [12]. With the HW/SW co-simulation feature, IDEA1 provides more detailed modeling of sensor node than NS-2. Benefiting from the efficient simulation kernel of SystemC and our optimized model implementation, the simulation speed of IDEA1 is much faster than that of NS-2.

IDEA1 models a sensor node exactly according to its hardware architecture. Each hardware component is modeled as an individual module. By doing this, the behaviors of each independent component can be

978-0-7695-4469-4/11 $26.00 © 2011 IEEE
DOI 10.1109/MASS.2011.95

accurately captured, which is the basis of accurate energy consumption prediction.

The network model of IDEA1 is inherited from the SystemC Network Simulation Library (SCNSL) [13] of alpha version which is a networked embedded system simulator. It includes 3 modules: node, node-proxy and network. During the initialization stage, each node registers its information (e.g., location, TX power and RX sensitivity) at a network class which maintains the network topology and transmits packets to other nodes. The node-proxy is an interface between the network and nodes. SCNSL demonstrates a great perspective for system-level simulation of WSN system, but it still has some limitations such as node-level simulation without any specific hardware platform specification or energy model. Many contributions considering the WSN specifications have been developed in IDEA1, which are summarized as follows.

- An energy model has been developed to enable the accurate energy consumption prediction. It considers the current consumption of not only each operation state of sensor nodes but also the transitions between different states.
- Many commercially available off-the-shell (COTS) processors and transceivers have been modeled, including ATMEL ATMega128, Microchip PIC16LF88, TI CC2420, TI CC1000 and Microchip MRF24J40. They are basic components of some COTS motes (e.g., MICA2 and MICAz).
- Many applications and one of the most-used WSN communication protocols, the IEEE 802.15.4 standard [14], have been implemented and analyzed.

The rest of this paper is organized as follows. Section II introduces the design and architecture of IDEA1. Section III validates its simulation results by some testbed measurements. Section IV evaluates its performance by comparing with NS-2. Section V concludes this paper.

II. DESIGN AND ARCHITECTURE OF IDEA1

A. *Architecture of IDEA1*

IDEA1 is a component-based simulation framework. Every component is modeled as an individual SystemC module communicating with each other via channels. The architecture of IDEA1 is illustrated in Fig. 1.

The node system is a complex model comprising 2 parts, hardware model and software model. The hardware components of a sensor node generally include a processing unit, a transceiver, several sensors and a battery. The software model consists of protocol stack and application implementations. All nodes are connected to a same network model via their proxies. At the initialization phase, every node registers its information (e.g., TX power and positions) at the network object. During simulation, the latter calculates the distance between the source and its destination, and forwards the packet according to the radio propagation

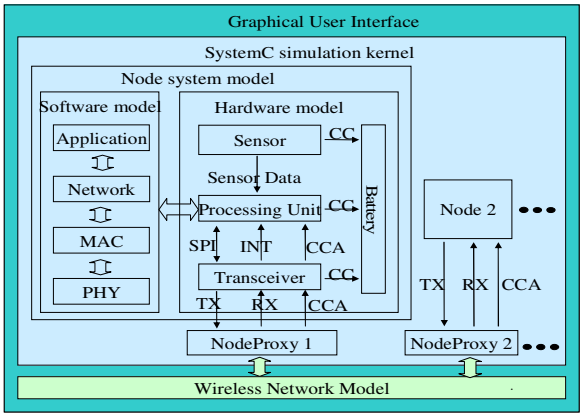

Figure 1. Architecture of IDEA1

models. If two packets arrive at a node simultaneously, a collision occurs. The SystemC kernel acts as the simulation engine. It schedules events and updates the states of modules every simulation cycle. All active processes are invoked sequentially at the same simulator time, which creates an illusion of concurrency.

For the node hardware model, the sensor is simulated as a stimuli generator providing analog sensor signals to the processing unit which converts this signal into digital format by a built-in Analog to Digital Convertor (ADC) and sends the data frame to the transceiver via a Serial Peripheral Interface (SPI) bus. The transceiver emits packets into network by different media access protocols. The transceiver reports the Clear Channel Assessment (CCA) results and some interrupts (e.g., receipt of a packet) to the processing unit.

The processing unit and transceiver are modeled as Finite State Machines (FSMs). During simulation, the state transition traces of each component are recorded. Each state is associated with a Current Consumption (CC) based on either experimental measurements or values in datasheets. The duration and current consumption of each transition between two states are also identified. Based on this information, the battery module calculates the energy consumption of each component and its residual capacity according to particular battery models during runtime.

A Graphical User Interface (GUI), as presented in Fig. 2, is developed to integrate all parts, which can facilitate the system configuration, network topology visualization, simulation control and result analysis. This GUI consists of three major parts: system configuration table (top-left) managing all the system parameters that users can set, network topology widget (top-right) showing the relative positions of all nodes and the radio connections among them, and a console (bottom) displaying the simulation log.

Users can configure many input parameters, such as node number, positions of nodes and various protocol parameters. They are defined in an eXtensible Markup Language (XML) file, which is read by the executable simulation program at the initialization stage. The

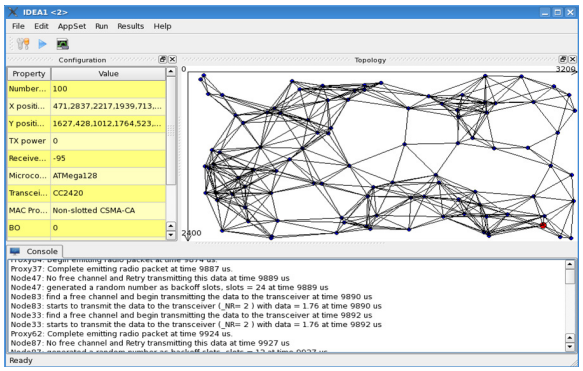

Figure 2. Graphical user interface of IDEA1

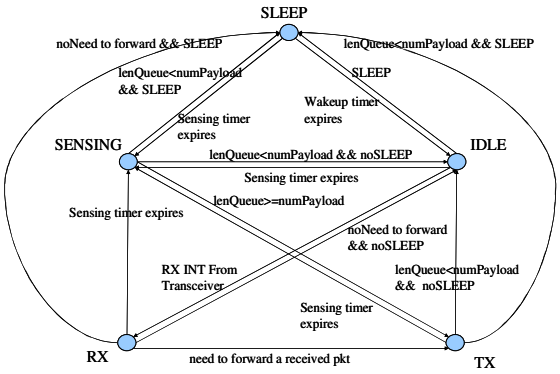

Figure 3. Microcontroller model of PIC16LF88

output results includes a value change dump file tracking the state transitions of some selected modules and a simulation log displaying all important steps and the final statistical results of network behaviors.

B. Hardware and Software Modeling

Many hardware components have been modeled in IDEA1. Due to the space constraint, only one node model is introduced to demonstrate the design process. The node prototype used in this paper is a sensor node developed in our laboratory, named as N@L (Node@Lyon). It is mainly composed of a PIC16LF88 microcontroller and a MRF24J40 transceiver. Its key feature is power efficient. The current consumption of active operation mode of PIC16LF88 is only 0.93-1.2mA [15]. Another feature is the hardware support of the IEEE 802.15.4 standard by MRF24J40.

The microcontroller communicates with transceiver via a SPI bus. To send a packet, the microcontroller writes the sensor data along with a MAC header into a TXFIFO of MRF24J40, which will add a PHY header and transmit the packet by using IEEE 802.15.4 protocols. When receiving, MRF24J40 verifies the cyclic redundancy check (CRC) and sends an interrupt to the microcontroller to report a receipt of packet. If the packet requires an acknowledgment (ACK), MRF24J40 will send an ACK automatically. The microcontroller and transceiver are modeled as finite state machines. As an example, the microcontroller model of PIC16LF88 is presented in Fig. 3.

In this model, the microcontroller is woken up periodically by a built-in timer to perform the sensing operation and try to transmit the data to its destination (a coordinator) if the data size is larger than a certain value (the payload field size of protocol-defined packet); otherwise, it may go to SLEEP or IDLE state depending on the application specifications. It will quit the TX state until the receipt of an end of transmission interrupt from the transceiver.

The FSM of microcontroller is controlled by software executions and interrupts generated by transceivers. The embedded software is divided into different tasks, such as data processing, ADC configuration and SPI communication. The execution time of each task is calculated according to their assembly codes. For example, the time taken by PIC16LF88 to complete an analog to digital conversion is 65.974 µs, including 11.974 µs acquisition time (Minimum Required Acquisition Time [15]) and 54 µs computation time (108 instructions with 8 MHz clock frequency) for the hardware configuration and result reading.

III. EXPERIMENTAL VALIDATION

In this section, a testbed is built to validate the simulation results of IDEA1. Two metrics are used to evaluate the network performance. They are defined as follows.

- Packet Delivery Rate (*PDR*): the ratio of the number of packets successfully received to the number of packets generated by nodes.
- Energy Consumption per Packet (*ECPkt*): the average energy consumed for successfully transmitting one packet.

A. Calibration of energy model

To calibrate the energy model, the current consumption of each operation mode of hardware components is measured on a single node. One resistor of 1 ohm was placed series with the power supply of a node (named as node0) in order to measure its current consumption. An instrumentation amplifier [16] with a gain of 76 is used to amplify the voltage across the resistor. A Tektronix MSO2012 mixed signal oscilloscope [17] is used to track current trace with the highest possible resolution. Tektronix MSO2012 provides a 1 GS/s sample rate. For the low current consumption of sleep mode, we use a digital multi-meter that can capture extremely small current.

A set of micro-benchmarks are developed to isolate the hardware consumption of microcontroller and transceiver to obtain the current consumptions of each component. The current consumptions of N@L motes are listed in Table I.

As shown in Table I, PIC16LF88 is a power-efficient microcontroller. It only consumes 1.386mA in the active mode. Both the microcontroller and transceiver need a period of time to wake up from the sleep mode.

TABLE I. CURRENT CONSUMPTIONS OF N@L MOTES

Microcontroller PIC16LF88		Transceiver MRF24J40	
Mode	**Consumption**	**Mode**	**Consumption**
Active	1.386mA	Sleep	17μA
SLEEP	7μA	RX	23.504mA
Sleep->active	7μA/1.846ms	TX(0dBm)	23.961mA
		TX(-10dBm)	22.901mA
		TX(-20dBm)	22.631mA
		TX(-30dBm)	22.409mA
		sleep->RX	6.7mA/720μs
		sleep->TX	6.7mA/720μs

B. Validation of simulation results

To validate the simulation results of IDEA1, a testbed based on N@L motes is established. It is a star topology consisting of eight nodes and one coordinator. The nodes send sensor data (one byte) to the coordinator periodically by using the IEEE 802.15.4 unslotted CSMA-CA algorithm. The parameters of this algorithm (e.g., *macMinBE*, *macMaxCSMABackoffs*, *macMaxFrameRetries*, etc.) are set as the default values defined in the IEEE 802.15.4 standard. The TX power of transceiver is set to 0 dBm. The nodes go to SLEEP mode after the transmission ends. They are woken up by a built-in timer. It is clocked by an external oscillator so as to continue to run during the sleep mode of microcontroller and generate an interrupt on overflow. The frequency of sensing operation is presented as sample rate. To set the node to different duty cycles, sample rate is set to 0.1, 1, 10, 100 and 1000 Hz. The ECPkt is measured by the current consumption trace of node0. PDR can be observed by the output trace of an I/O pin of coordinator, which is recorded by the oscilloscope. Once the coordinator receives a packet from node0 with different sequence number, it toggles one of its I/O pins. The application performed by the testbed has also been implemented in IDEA1 with the same system configuration. The simulation and measurement results are presented in Fig. 4 and 5.

The average deviations between IDEA1 simulations and testbed measurements are 5.2% and 6.5% for PDR and ECPkt respectively. Therefore, the average deviation for the two metrics is 5.9% which can be accepted for general system-level simulations.

With a small sample rate (0.1, 1, 10 and 31.25), the system is light loaded and every node can finish its transmission before a new sensing operation; thus the PDRs remain stable. Since the average number of successful transmitted packets per sample interval is almost the same for different sample intervals, a bigger sample interval comprises longer period of sleep mode and its ECPkt is larger. The largest sample rate without transmission overlapping is 31.25 Hz.

When the sample rates are 100 or 1000 Hz, the PDRs decrease due to the increase of collisions, and the ECPkts augment because of less number of packets that successfully received by the coordinator.

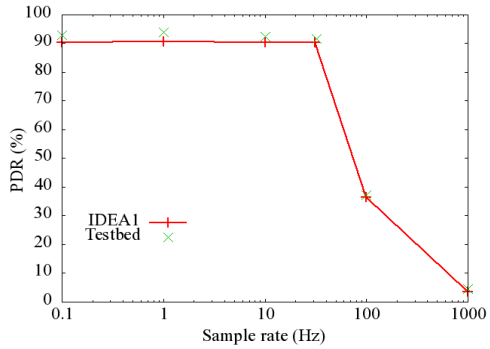

Figure 4. Measured and simulated *PDR* results

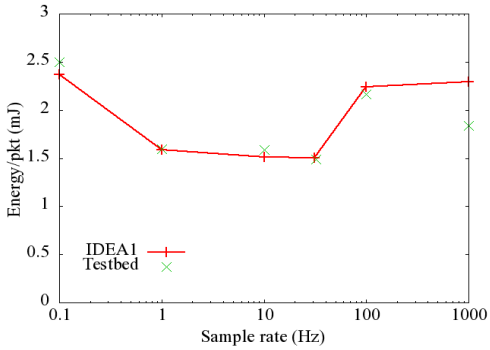

Figure 5. Measured and simulated *ECPkt* results

IV. PERFORMANCE EVALUATION OF IDEA1

In this section, the performances of IDEA1 are compared to NS-2 in the aspects of accuracy and simulation time.

A. Model implementations

The same application of section IIIB is studied by NS-2 and IDEA1. The NS-2 model used in this paper is based on an existing IEEE 802.15.4 NS-2 model in release 2.34 [4]. The model has been modified significantly, since it was built complying with an earlier standard edition (IEEE 802.15.4 draft D18), which has been replaced by the latest revised release IEEE Std 802.15.4-2006. The extensions provided in [5] have also been added, such as sleep mode and symbol period CCA duration implementation.

The hardware prototype used for this application is N@L motes and the energy model is calibrated according to the testbed measurements presented in section IIIA. The nodes use slotted IEEE 802.15.4 CSMA-CA algorithm to access channel. With this algorithm, a coordinator sends beacon packets periodically to synchronize the attached nodes and describe a superframe structure, as shown in Fig. 6.

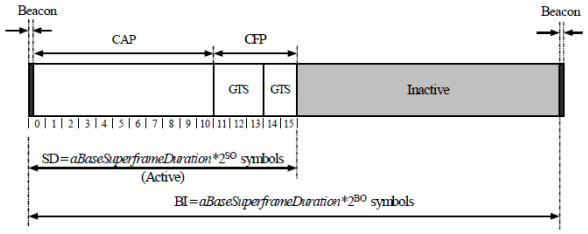

Figure 6. IEEE 802.15.4 superframe structure [14]

Beacon Interval (BI) defines the superframe length including an active period and, optionally, an inactive period. The active portion consists of two periods, namely contention access period (CAP) and contention free period (CFP). During CAP, nodes use the slotted CSMA-CA algorithm to access the channel. During CFP, many guaranteed time slots can be allocated to different nodes so as to allow them operating on a single channel exclusively. Superframe Duration (SD) presents the length of active period. BI and SD are determined by two parameters respectively, Beacon Order (BO) and Superframe Order (SO). The minimum duration of a superframe (*aBaseSuperframeDuration*) is 15.36 ms if the data rate is 250 kbps.

Many cases with various configurations of parameters (mainly BO, SO and sample rates) have been studied. To investigate the effect of SO and BO, SO is fixed to 0 and BO is set to 0, 1 and 2, which results in a constant active period of 15.36 ms and a superframe of 15.36, 30.72 and 61.44 ms respectively. Other parameters are set to the values defined by default in the IEEE 802.15.4 standard. Each simulation includes 10000 samples, for example, when sample rate is 0.1, the application lasts 27.8 hours. Each case is simulated 100 times with different seeds for the generators of random backoff slot number.

B. Simulation Results

Three types of simulation results are compared, including NS-2, IDEA1 with hardware implementations (denoted as IDEA1_HW) and IDEA1 without hardware information (IDEA1_NOHW). In the last kind of simulations, all the timing parameters about the hardware operations are set to 0, thus they do not consume any time or energy. The simulation results are presented in Fig. 7 and 8.

The deviations between IDEA1_HW and NS-2 of the two metrics, PDR and ECPkt, are 2.7% and 49.3% respectively. The ones between IDEA1_NOHW and NS-2 are 1.0% and 8.3%. Therefore, the average deviation between IDEA1_HW and NS-2 is 26%, and the one between IDEA1_NOHW and NS-2 is 4.65%. The former is bigger since more detailed information of HW/SW operations has been considered. For example, as shown in Fig. 8, when the sample rate is 0.1, the deviation of ECPkt results between IDEA1 and NS-2 is very large, because the SPI communications of microcontroller and transceiver account takes 42.4% of the power consumption of microcontroller.

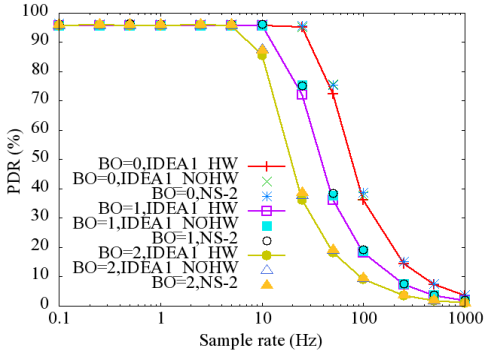

Figure 7. *PDR* simulation results of IDEA1 and NS-2

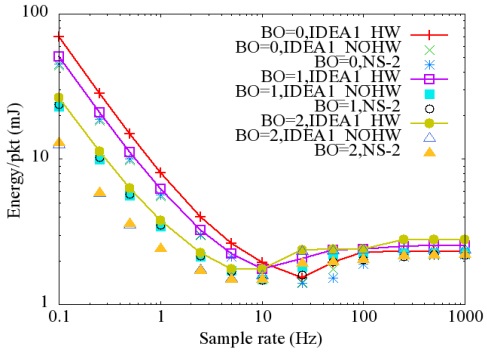

Figure 8. *ECPkt* simulation results of IDEA1 and NS-2

The simulation results proved that IDEA1 provides more accurate modeling of real WSN system. In the rest of this section, the phenomena illustrated in Fig. 7 and 8 are briefly explained. As the sample rate increases from 0.1 to 1000, the system goes through 3 different stages, i.e., lightly loaded, heavily loaded and saturated.

If the sample rates are small, the system is lightly loaded. During this stage, the sample interval is long enough for every node to accomplish its transmission before the next sensing operation and the average numbers of transmitted packets during one sample interval are same for different sample rates; thus the PDRs remain stable. The ECPkts decrease as the sample interval becomes shorter. For a fixed sample rate, the smallest BO consumes the most energy since one sample interval includes more superframes and the nodes have to wake up to track the beacon packet at the beginning of each superframe.

As the sample rate increases, the number of sensor data need to be sent per unit time augments and PDR begins to decrease due to the increase of collisions. The PDRs with bigger BO begin to decrease first, because SO is the same and a bigger BO means that one sample interval includes less number of active portions. The smallest energy consumption occurs at the beginning of the heavily loaded stage. In this case, every node can accomplish its transmission before new sensor data arrives, but the interval between the last node turns to sleep and the next sensor data arrives is so short that the nodes spend the least energy in sleep mode. As the sample rate continues to increase, the energy

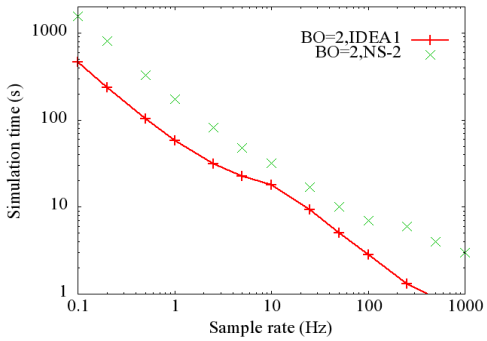

Figure 9. Simulation time of NS-2 and IDEA1

consumption augments due to the increase of collisions and the successfully transmitted packets decrease.

When the system becomes completely saturated, nodes always have several pending data to send. During this stage, the number of successfully transmitted packets per superframe is same because the same length of active portion. Therefore, The ECPkts for the same BO remain constant and the ECPkts with a bigger BO are larger.

C. *Simulation Time*

For the simulations of section IV, the speed of IDEA1 is about 2 times faster than NS-2. The simulation time of NS-2 and IDEA1_NOHW for the application with BO set to 2 is presented in Fig. 9.

All the simulations are executed individually on a server with an Intel 2.66 GHz Xeon X3230 processor and a 4.6 GB memory. For the application lasting 27.8 hours with a sample rate of 0.1 Hz, the simulation time of IDEA1 and NS-2 are 7.35 and 24.0 minutes respectively. The high speed simulation of IDEA1 profits mainly from the efficient simulation kernel of SystemC and our optimized model implementation. SystemC provides a wait mechanism which can set relative processes to inactive state until an interesting event occurs. In our model, this event interrupt method is used in the FSM implementation. Instead of checking the states of microcontroller and transceiver every simulation cycle, their FSMs are woken up only when an interesting event occurs, which reduces the simulation time significantly.

V. CONCLUSIONS

This paper presented IDEA1, a validated system-level simulator for WSN. It enables the design space exploration at an early stage. It models the sensor node in SystemC, which makes the simulation to be a part of the HW/SW design of sensor nodes. It supports a modular design of sensor nodes and WSN applications. Many hardware platforms have been modeled and the IEEE 802.15.4 standard has been implemented. The average deviation between the IDEA1 simulations and the experimental measurements is 5.9%. IDEA1 can provide more detailed modeling of sensor network than NS-2 but with less simulation time.

REFERENCES

[1] W. Du, D. Navarro, F. Mieyeville, and F. Gaffiot, “Towards a Taxonomy of Simulation Tools for Wireless Sensor Networks,” Proc. International Conference on Simulation Tools and Techniques (SIMUTools 2010), Mar. 2010.

[2] “Network simulator ns-2”, http://www.isi.edu/nsnam/ns/, Feb. 2010.

[3] A. Varga, “The OMNeT++ discrete event simulation system,” Proc. European Simulation Multiconference, Jun. 2001, pp. 319-324.

[4] J. Zheng and M. J. Lee, “Will IEEE 802.15.4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard,” Communications Magazine, IEEE, vol. 42, no. 6, pp. 140– 146, Jun. 2004.

[5] I. Ramachandran, A. K. Das, and S. Roy, “Analysis of the contention access period of ieee 802.15.4 mac,” ACM Trans. Sen. Netw., vol. 3, March 2007.

[6] P. Levis, N. Lee, M. Welsh, and D. Culler, “Tossim: Accurate and scalable simulation of entire tinyos applications,” Proc. International Conference on Embedded Networked Sensor Systems, Nov. 2003, pp. 126-137, doi: 10.1145/958491.958506

[7] B. Titzer, D. Lee, and J. Palsberg, “Avrora: Scalable sensor network simulation with precise timing,” Proc. International Symposium on Information Processing in Sensor Networks, April 2005, pp. 477-482, doi: 10.1109/IPSN.2005.1440978

[8] IEEE Std 1666 - 2005 IEEE Standard SystemC Language Reference Manual. IEEE Std 1666-2005, pages 1–423, 2006.

[9] G. Sachdeva, R. D¨omer, and P. Chou, “System modeling: A case study on a wireless sensor network,” University of California, Irvine, Tech. Rep. CECS-TR-05-12, 2005.

[10] J. Hiner, A. Shenoy, R. Lysecky, S. Lysecky, and A. G. Ross, “Transaction-level modeling for sensor networks using SystemC,” in Proc. of the 2010 IEEE int. conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing, ser. SUTC ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 197–204.

[11] M. Damm, J. Moreno, J. Haase, and C. Grimm, “Using transaction level modeling techniques for wireless sensor network simulation,” in Proc. of the Conference on Design, Automation and Test in Europe, ser. DATE ’10, 2010, pp. 1047–1052.

[12] S. Kurkowski, T. Camp, and M. Colagrosso, “Manet simulation studies: the incredibles,” SIGMOBILE Mob. Comput. Commun. Rev., vol. 9, pp. 50–61, October 2005.

[13] F. Fummi, D. Quaglia and F. Stefanni, “A SystemC-based Framework for Modeling and Simulation of Networked Embedded Systems,” Proc. Forum on Specification, Verification and Design Languages (FDL2008), Sep. 2008, pp. 49-54, doi:10.1109/FDL.2008.4641420

[14] IEEE. Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPANs). IEEE Sandard 802.15.4-2006

[15] Microchip Technology Inc. “Picmicro mid-range mcu family reference manual.”, ww1.microchip.com/downloads/en/devicedoc/33023a.pdf, Feb. 2011.

[16] Linear Technology, LT1014 amplifier, http://cds.linear.com/docs/Datasheet/10134fd.pdf, Feb. 2011.

[17] Tektronix MSO2012 mixed signal oscilloscope, http://www.tek.com/, Feb. 2011.