IDEA1: A validated SystemC-based system-level design and simulation environment for wireless sensor networks

Wan Du, Fabien Mieyeville, David Navarro, Ian Connor

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

IDEA1: A validated SystemC-based system-level design and simulation environment for wireless sensor networks

Wan Du*, Fabien Mieyeville, David Navarro and Ian O Connor

Abstract

This article presents IDEA1, a SystemC-based system-level design and simulation framework for WSNs. It allows the performance evaluation (e.g., packet delivery rate, transmission latency and energy consumption) at high level, but with elaborate models of the hardware and software of sensor nodes. Many hardware components are modeled and the IEEE 802.15.4 standard is implemented. IDEA1 uses a clock-based synchronization mechanism to support simulations with cycle accurate communication and approximate time computation. The simulation results have been validated by a testbed of 9 nodes. The average deviation between the IDEA1 simulations and experimental measurements is 4.6%. The performances of IDEA1 have also been compared with NS-2. 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. Finally, IDEA1 is used to study a real-time industrial application in which a wireless sensor and actuator network is deployed on a vehicle to measure and control vibrations. By the simulation, some preliminary designs based on IEEE 802.15.4 protocols and two different hardware platforms are evaluated.

1 Introduction

In recent years, numerous applications of wireless sensor networks (WSNs) have been developed. Different applications have diverse requirements; for example, a real-time industrial application requires short packet delivery latency, but a lifetime of weeks is often enough. In contrast, a remote environment monitoring system prefers a long lifetime of years with a low duty cycle. To meet the diversity of these requirements, designers need to consider a great number of node-level design choices (e.g., energy consumption of hardware components and processing capability) and many protocol-level parameters (e.g., anti-collision algorithms and routing approaches). Simulation is a cheap and quick way to perform many experiments with different hardware prototypes and network settings [1]; thus, a simulation tool is needed to explore the huge design space at an early stage before devoting too much time and resources.

The requirements of small size and low cost result in limited energy supply on sensor nodes. In order to extend the network lifetime, many efforts have been taken to reduce the energy consumptions of hardware, software, communication protocols and applications. Therefore, it is necessary to accurately predict the energy consumption of WSN, which requires detailed models of the hardware and software (HW/SW) of sensor nodes.

Many simulation tools for WSN have been developed by using different methodologies such as general-purpose network simulation, operating system (OS) emulation, instruction set simulation and System-Level Description Language (SLDL). However, most of them are implemented in general programming languages such as C++ and Java that do not support directly the HW/SW co-simulation. Only a few simulators designed in SLDLs provide native support to model concurrency, pipelining, structural hierarchy, interrupts and synchronization primitives of embedded systems [2].

As an SLDL, SystemC is a C++ class library for system and hardware design [3]. It can model the embedded system at different abstraction level and allow designers to focus on the system functionalities by hiding the unnecessary details of communication and computation. Four SystemC-based WSN simulators [4-7] have been

* Correspondence: wan.du@ec-lyon.fr
Lyon Institute of Nanotechnology, University of Lyon, Lyon, France

developed; however, none of them has been validated with experimental measurements or evaluated comprehensively by comparing with other simulators.

To exceed this limitation, a novel SystemC-based WSN simulator named IDEA1 (hIerarchical DEsign plAtform for Wsn and Architectural Node exploration) is developed. A testbed of 9 sensor nodes has been built to validate the simulation results of IDEA1. The deviation of IDEA1 simulations and the experimental measurements is small enough to be 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 [8]. With the HW/SW co-simulation feature, IDEA1 provides more detailed information of sensor node operations than NS-2, which can help designers to better analyze the energy dissipation of networks. Benefiting from the efficient simulation kernel of SystemC and our optimized model implementation, the simulation speed of IDEA1 is much faster than NS-2.

IDEA1 allows rapid performance evaluation at system level. The simulation results include packet delivery rate, transmission latency and power consumption. Many commercial off-the-shelf (COTS) hardware components, such as MICAz and MICA2, are modeled. The IEEE 802.15.4 standard [9] is implemented. It has been widely utilized in WSN applications since it is designed for low data rate, short distance and low-power-consumption applications in conformity with the constraints of WSN systems [10].

One important feature of IDEA1 is the accurate prediction of energy consumption of each sensor node and the whole network. It implements a clock-based synchronization mechanism to provide performance evaluation with cycle accurate communication and approximate time computation as a bus-functional model defined in [11]. The energy model implemented in IDEA1 takes into account the power consumptions of all operation modes of each hardware component and transitions between different modes.

This paper is organized as follows. Section 2 summarizes the related works and the position of IDEA1 with respect to other simulators. Section 3 introduces the design and architecture of IDEA1. Section 4 validates its simulation results by some testbed measurements and evaluates its performance by comparing with NS-2. Section 5 demonstrates its usability and design flow by studying a real industrial application of WSN in vibration control. Section 6 concludes this paper and introduces the future work.

2 Related work

At present, WSN simulations mainly involve two parts: node system modeling and network modeling. The node system includes the hardware and software of a sensor node; the network modeling handles the interconnections among nodes. According to the key techniques they adopt, the existing WSN simulators can be divided into 3 categories: network simulators, node emulators and node simulators. Network simulators refer to the general-purpose network simulators that have been applied to WSN simulations. They emphasize on the network modeling and are enhanced with WSN-specific node models. Node emulators principally focus on the modeling of embedded software execution. They include the operating system emulators and instruction set simulators (ISS) of the processing unit on sensor nodes. Node simulators are generally developed in SLDLs which can provide behavioral models of sensor nodes and are compatible to the design flow of embedded systems.

A more detailed analysis of the simulation and recent simulators for WSN can be found in our previous work [12]. In this paper, we focus on specifying the distinguished features of IDEA1 while introducing many typical simulators in each category.

2.1 Network simulators

Many general-purpose network simulators, such as NS-2 [13] and OMNeT++ [14], have been utilized in WSN simulations. NS-2 is a discrete event, object-oriented simulator. SensorSim [15] is the first contribution for NS-2 to WSN simulation, where an 802.11 network is modeled with considerations of power models of hardware components. However, it has the deficiency of lacking a CPU model, and IEEE 802.11 is not widely adopted in WSN applications because of the high power consumption. An NS-2 IEEE 802.15.4 model is proposed in [16], while an energy model is added in [17]. However, NS-2 does not scale well in terms of memory usage and simulation time [18]. OMNeT++ adopts component-based programming model. An OMNeT++ IEEE 802.15.4 model is implemented in [19]. The performance of IEEE 802.15.4 standard in the context of cyber-physical systems has been evaluated. PAWiS [20] is an OMNeT++ based WSN simulator that features the power consumption estimation.

Besides the extensions to general-purpose network simulators, some WSN-specific network simulators have also been developed. NetTopo [21] is an integrated simulator that provides the simulation of virtual WSN and the visualization of real testbeds. It also supports the interaction between the simulated WSN and real testbeds.

Generally, the network simulators specialize in network modeling and support a complete protocol stack. They have the advantages of extensibility, heterogeneity support and easy to use. However, the simulation

models implemented in these simulators are difficult to be reused in system design or executed on real nodes; moreover, the energy consumption estimation is usually based on some simple assumptions of the sensor node operations; for example, the processor and radio-frequency (RF) transceiver have same operating state, but in fact the processor can be in sleep mode when the transceiver is listening to channels. IDEA1 overcomes these drawbacks by component-based hardware modeling of sensor nodes. Every hardware component is modeled as an individual module which operates independently. SystemC-based IDEA1 is not only a simulator but also a system design environment for WSN. Having a sensor node model, it is possible to evaluate its network performance. Once the requirements of final system are met, the real implementation of HW/SW components can start from this description.

2.2 Node emulators

TOSSIM [22] is an operating system emulator designed for the execution of TinyOS applications. It offers a controlled environment facilitating the development of algorithms and the study of system behaviors. ATEMU [23] is an instruction-level cycle accurate emulator written in C which has an ISS of AVR processor as the simulation kernel. It also supports other peripheral devices on MICA2 mote like the radio. Avrora [24] improves the performance of ATEMU in scalability.

Node emulators provide the highest behavioral and timing accuracy of software execution. The embedded software developed for physical platforms can be executed directly in the simulation framework with little or no modifications. However, they are generally constrained to specific predefined hardware platforms or operating systems. Due to the system-level abstraction, IDEA1 can quickly model the behaviors of an application based on a certain hardware platform at different timing-accurate levels, which accelerates the model development and simulation speed.

2.3 Node simulators

Wireless sensor network simulator (WISENES) [25] is developed in Specification and Description Language (SDL) which is a high-level abstraction language widely used in communication protocol design. The key feature of WISENES is that its simulation models are reusable in system design. However, it only contributes to the software implementation, but not the HW/SW co-design.

Kashif Virk et al. [5] have developed a SystemC-based modeling framework for WSN. It models the applications, real-time operating systems, sensors, processor and transceiver at node level and signal propagations at network level. However, only a behavior waveform of media access control (MAC) layer (states of the sending and receiving tasks) has been presented in [5]. The SNOPS framework [7] is a transaction-level modeling (TLM)-based WSN simulator. A sensor node transmits or receives a data packet to or from an environment model by transaction exchanges. In [7], it is proved that the SNOPS framework requires 49.7% less simulation time than PAWiS [20]. ATLeS-SN (Arizona Transaction-Level Simulator for Sensor Network) [6] is another TLM-based sensor network simulation environment developed in SystemC. It models a sensor node in 3 components: application specification, network stack implementation and sensor system. The physical channel is modeled as a component. ATLeS-SN demonstrated the feasibility of using TLM for sensor network application, but no standard networking protocol has been implemented.

SystemC Network Simulation Library (SCNSL) [4] is a networked embedded system simulator, written in SystemC and C++. It includes 3 modules: node (SystemC), node-proxy (SystemC) and network (C++). 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. By using node-proxy, nodes can be designed as pure SystemC modules so as to exploit all advantages of SystemC in HW/SW co-design and verification. 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 or energy model.

IDEA1 is based on the SCNSL library of alpha version. The network model of IDEA1 is inherited from SCNSL; however, many contributions have been developed, which are summarized as follows.

- Emphasizing the modular design, but not like ATLeS-SN, 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 hardware components can be accurately captured, which is the basis of energy consumption estimation. Many COTS processors and transceivers have been modeled, including AT-MEL ATMega128, Microchip PIC16LF88, TI CC2420, TI CC1000 and Microchip MRF24J40. They are basic components of some COTS motes (e.g., MICA2 and MICAz).
- The software, such as applications and protocols, is implemented in separated modules which can control the operations of processor. Many applications and one of the most used WSN communication

protocols, the IEEE 802.15.4 standard, have been implemented.
- An energy model has been developed to enable the accurate energy consumption prediction. It has been calibrated by some experimental measurements.
- A graphical user interface (GUI) has been developed to facilitate the system configuration, the observation of network topology and the analysis of simulation results.
- The simulation results of IDEA1 have been validated by a testbed consisting of 9 nodes.
- The performances of IDEA1 have also been compared with NS-2.

3 Design and architecture of IDEA1

In this section, the design and architecture of IDEA1 are presented, including the design framework, HW/SW modeling, energy model and user interface.

3.1 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 Figure 1.

The node system is a complex model comprising two 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 object via their proxies. At the initialization phase, every node registers its information at the network module. During simulation, the network object calculates the distance between the source and its destination and forwards the packet according to the radio propagation 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. A GUI based on Qt platform [26] is developed to integrate all parts, which can facilitate the system configuration, network topology visualization, simulation control and result analysis.

In the node hardware model, the sensor is simulated as a stimuli generator that is an interface specifying how the physical environmental parameters vary in spatial and temporal terms. The processing unit converts the

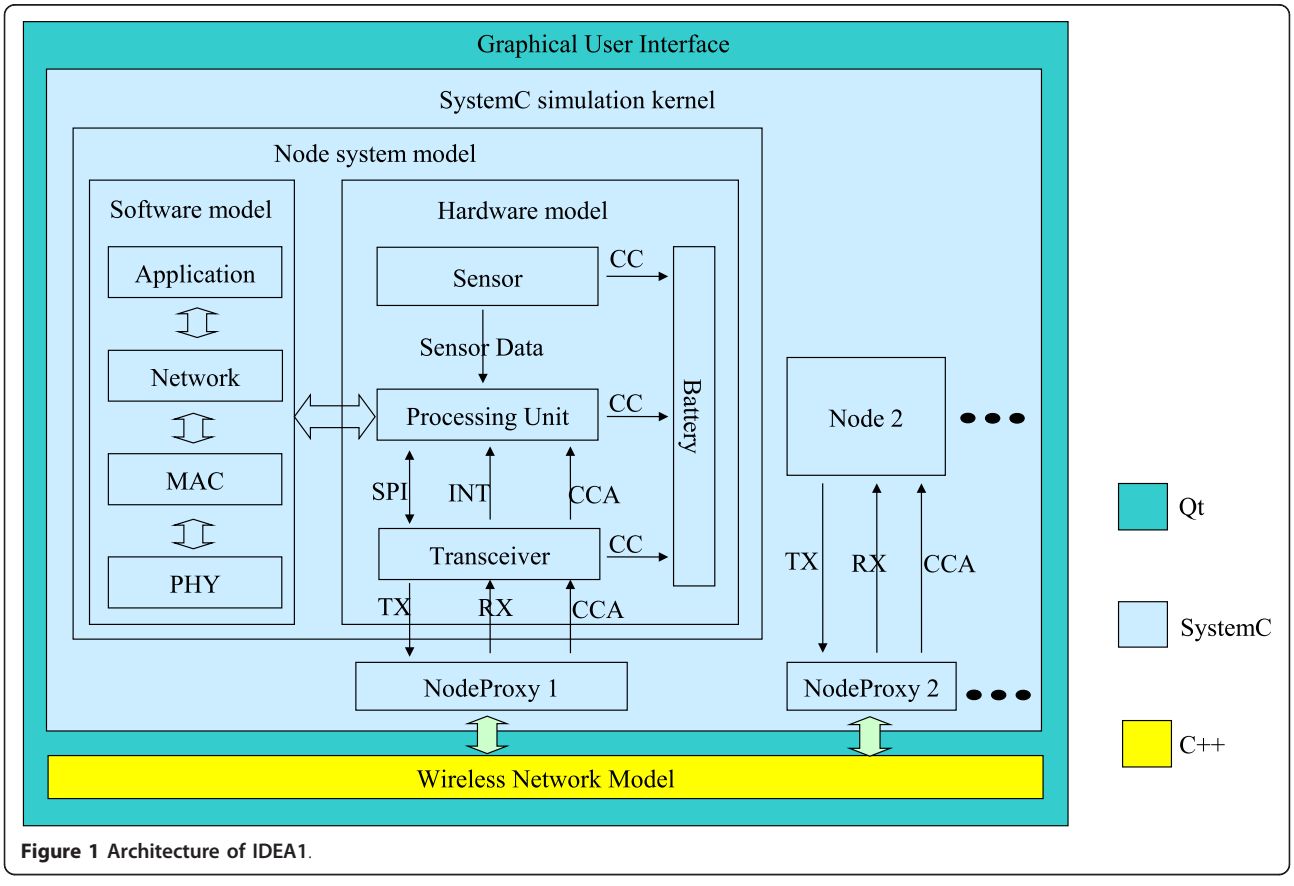

Figure 1 Architecture of IDEA1.

analog signal generated by the sensor module 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) result 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.

3.2 Design framework of IDEA1

The design framework of IDEA1 is presented in Figure 2. The input parameters of IDEA1 are defined in an eXtensible Markup Language (XML) file, which is read by the executable simulation program at the beginning of simulation. Application parameters describe the network compositions and application tasks. Network parameters define the impact of environment on radio propagations. The protocol can be tuned by setting the protocol parameters. Node, microcontroller and transceiver parameters specify the capabilities of sensor node platforms and some behaviors of hardware components. The output results include simulation log that displays all important steps of network behaviors, a value change dump (VCD) file that tracks the state transitions of some selected modules, and the network topology. The statistical results of network behaviors are provided at the end of simulation log. They can be divided into three categories, i.e., packet delivery, latency and energy consumption.

3.3 Hardware and software modeling of microcontroller

Many COTS hardware components have been modeled in IDEA1. To introduce the design process, as an example, 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 an MRF24J40 transceiver. Its key feature is power efficient. The current consumption of active operation mode of PIC16LF88 is only 0.93-1.2mA [27]. Another feature is hardware support of IEEE 802.15.4 standard by MRF24J40.

The microcontroller communicates with the transceiver via a SPI bus. To send a packet, the microcontroller needs to write the sensor data along with a MAC header to MRF24J40, which will add a synchronization header, PHY header and frame check sequence (FCS) 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 is modeled as a finite state machine, as presented in Figure 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). When in the SENSING state, it performs the sensing operating which is modeled by data generation in the sensor module and analog to digital conversion in the microcontroller module. After conversion, the microcontroller stores the sensor data in a buffer. It will go to either SLEEP or IDLE state depending on the application specifications if the data size is less than a certain value (the payload field size of protocol-defined packet); otherwise, it will go to TX state and send the sensor data to the transceiver via SPI. It will quit the TX state until the transmission finishes. After a transmission, the microcontroller may stay in TX state and transmit another packet, or it will go to SLEEP or IDLE state. When in IDLE state, if a packet is received, it will be informed by an interrupt from transceiver and go to RX state. If the packet is intended to other nodes, the microcontroller needs to forward it to its destination.

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 54 μs computation time (108 instructions with 8 MHz clock frequency) and 11.974 μs acquisition time (minimum required acquisition time [27]). The computation time includes the hardware configuration and result reading.

3.4 RF transceiver modeling

MRF24J40 provides full hardware supports of the IEEE 802.15.4 standard. The three IEEE 802.15.4 media access algorithms have been modeled, including unslotted carrier sense multiple access with collision avoidance (CSMA-CA), slotted CSMA-CA and guaranteed time slots (GTS). Three finite state machines of transceiver are developed according to these MAC algorithms. Due to the limit of space, only the model of the most

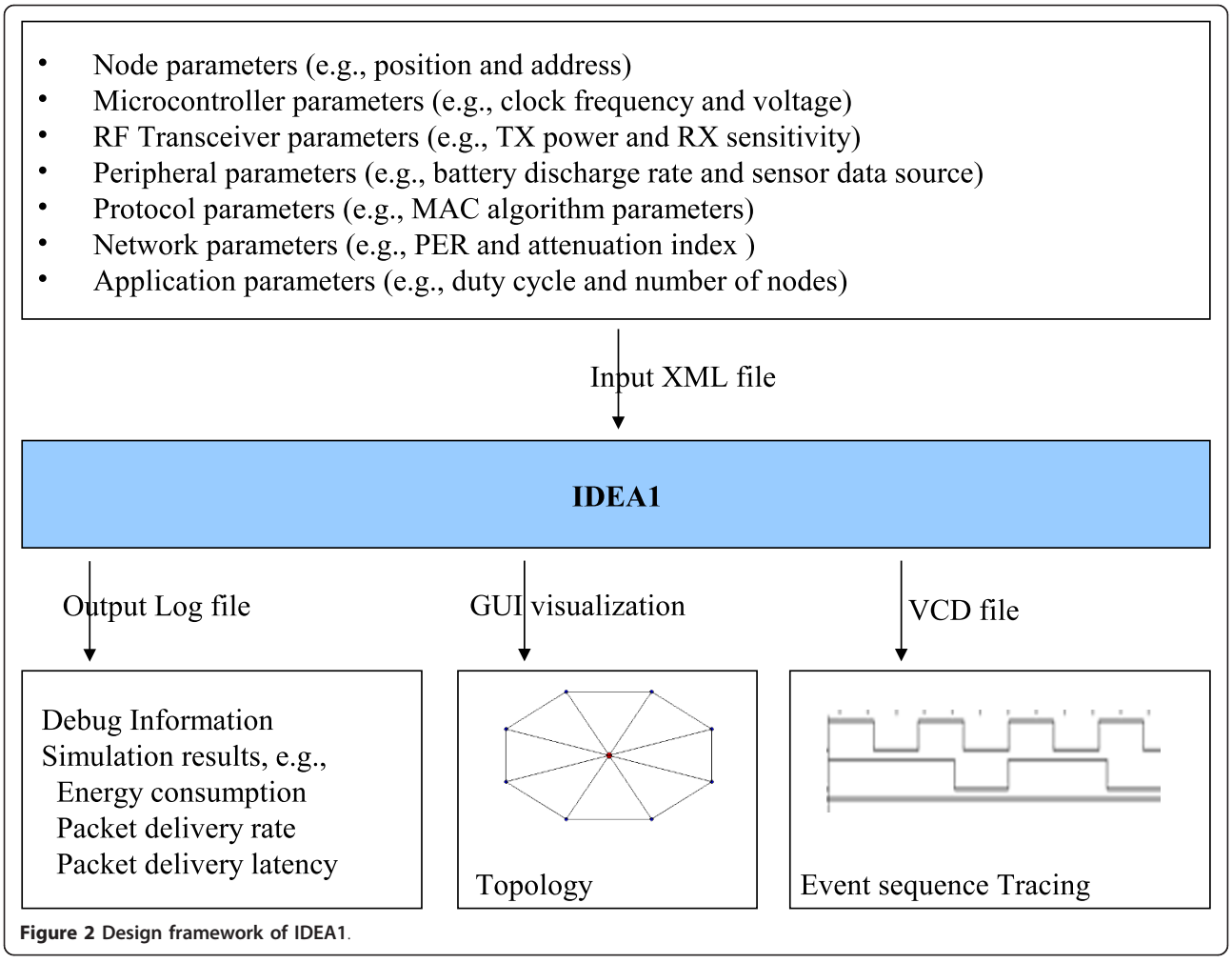

Figure 2 Design framework of IDEA1.

complex algorithm, slotted CSMA-CA, is presented. In this section, we first briefly introduce some important conceptions of IEEE 802.15.4. Secondly, the design of transceiver models is described.

IEEE 802.15.4 supports two operation modes: nonbeacon-enabled and beacon-enabled. If nonbeacon mode is enabled, nodes perform independently and they transmit data by using unslotted CSMA algorithm. With the beacon-enabled model, a coordinator sends beacon packets periodically to synchronize the attached nodes and describe the superframe structure. One superframe includes 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 GTSs (up to 7) can be allocated, which allow the node to operate on a channel that is dedicated exclusively to it. Beacon interval (*BI*) defines the superframe length and 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).

$$BI = aBaseSuper\ frameDuration \cdot 2^{BO} \quad (1)$$

$$SD = aBaseSuper\ frameDuration \cdot 2^{SO} \quad (2)$$

The minimum duration of a superframe (*aBaseSuperframe Duration*) is 960 symbols corresponding to 15.36 ms if the data rate is 250 kbps.

The state transition of transceiver is triggered by three events, i.e., the protocol algorithms, microcontroller commands or network events. The model of MRF24J40 with slotted CSMA-CA algorithm is presented in Figure 4.

If it is a coordinator, the BEACON ACQUIREMENT/TRANSMISSION state is to broadcast a beacon packet; for a device node, it is BEACON ACQUIREMENT. On a successful receipt of beacon packet, the node may go to RX state if it has no data to send, or it continue with the transmission in the previous superframe. The

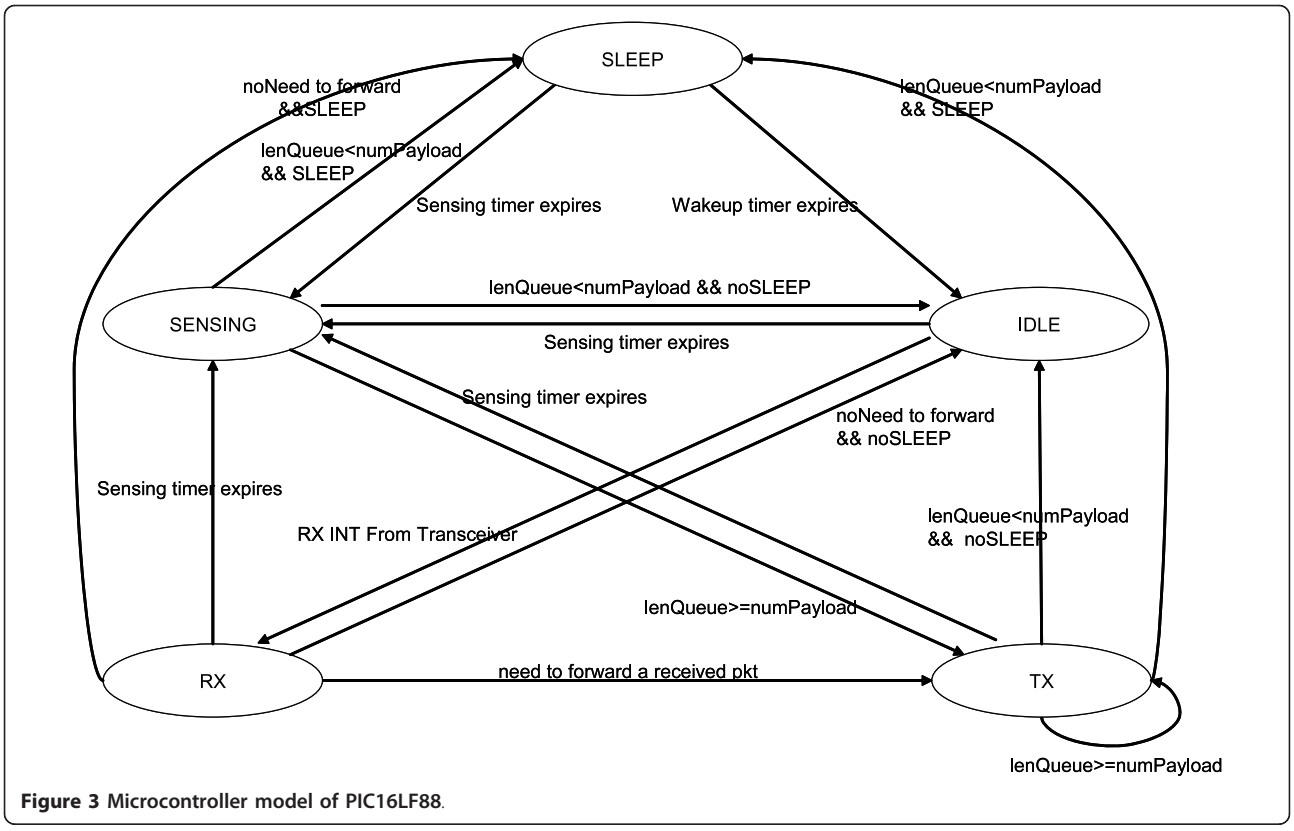

Figure 3 Microcontroller model of PIC16LF88.

process of slotted CMA-CA algorithm is similar to the unslotted algorithm except the backoff period boundaries of every node should be aligned with the superframe slot boundaries and the MAC sublayer should ensure that the PHY commences all of its transmissions on the boundary of a backoff period. One backoff period includes 20 symbols.

For the CSMA-CA algorithms, firstly, the number of backoff (*NB*) is initialized to 0. Then, the algorithm starts counting down a random number of backoff periods. When the timer of backoff expires, the algorithm performs channel assessment. If the channel is idle, the node starts transmitting; otherwise, *NB* is increased. If *NB* does not reach the maximum number of backoff (*macMaxCSMABackoff*), the algorithm backoffs again; otherwise, the channel access operation fails.

3.5 Energy model

Each state of the main hardware components in a sensor node is associated with a current load. The duration and current consumption of each transition between two states are also identified. During the simulation, the states of hardware components are updated according to the software execution and network events. The energy consumed by node i can be calculated as follows.

$$E_i = \sum_{j=0}^{N} \left(\sum_{k=0}^{M} E_{ijk} + \sum_{l=0}^{O} E_{ijl} \right) = \sum_{j=0}^{N} \left(\sum_{k=0}^{M} V \cdot I_{ijk} \cdot t_{ijk} + \sum_{l=0}^{O} V \cdot I_{ijl} \cdot t_{ijl} \right) \quad (3)$$

where E_{ijk} presents the energy consumption of the kth state of the jth component of node i, and E_{ijl} presents the energy consumption of the lth state transition of the jth component of node i. The node has N components consuming energy. Each component has M states and O transitions. During the simulation, the state transition traces of each component are recorded; thus, the time spent on different states and transitions, t_{ijk} and t_{ijl}, are known. Based on this information, the battery module calculates the energy consumptions of each component as well as the network lifetime.

3.6 Graphical user interface

A GUI is developed to visualize the network topology and help users who are not familiar with SystemC to do some experiments on IDEA1. It is designed as a plug-in to the simulation environment so that the experienced designers can also write SystemC code directly to configure applications and control simulations. An example of IDEA1 graphical user interface is presented in Figure 5.

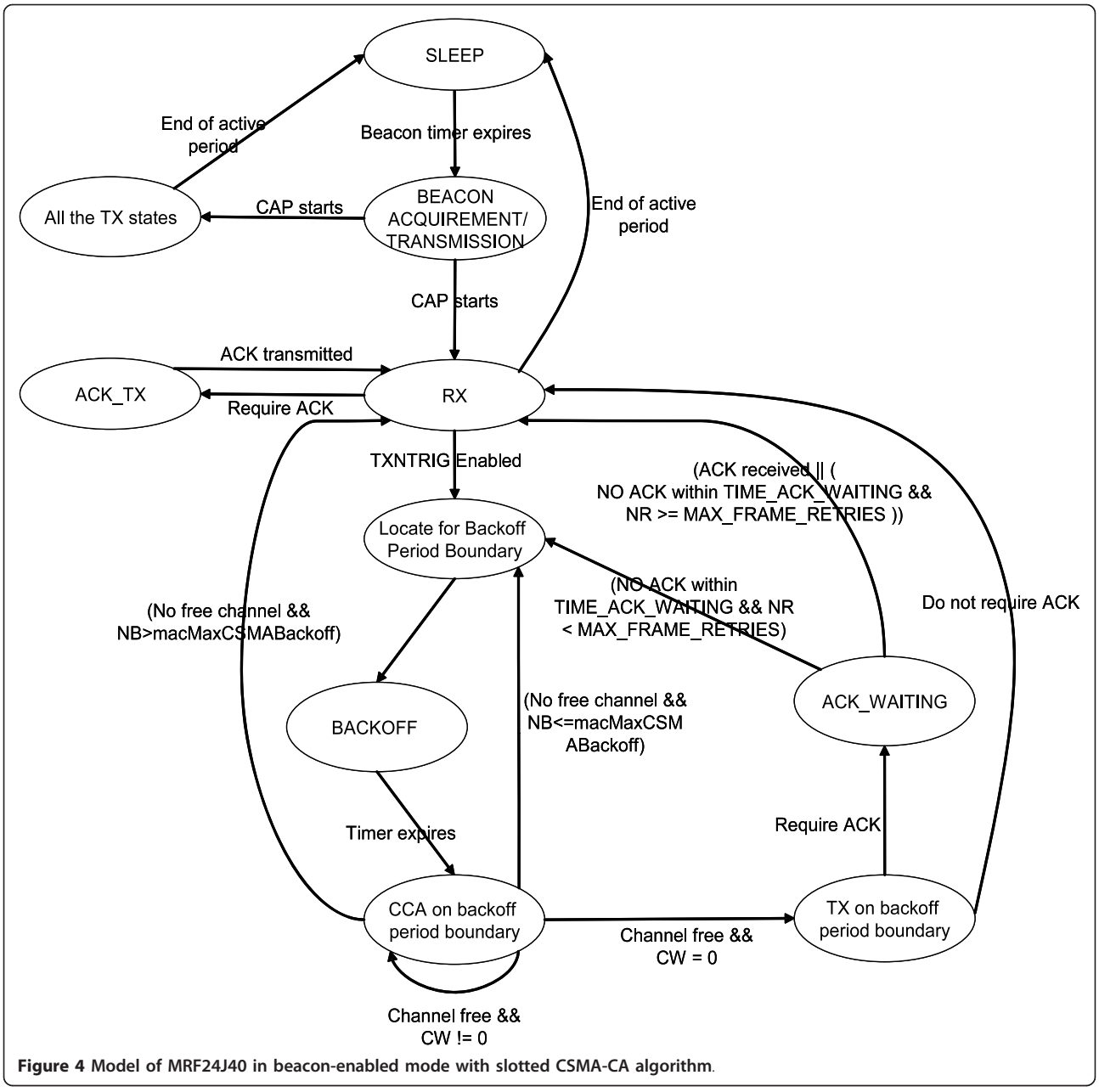

Figure 4 Model of MRF24J40 in beacon-enabled mode with slotted CSMA-CA algorithm.

The GUI illustrated in Figure 5 consists of three major parts: system configuration table (top-left of the main window) managing all the system parameters, 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.

4 Evaluation

In this section, firstly, a testbed is built to validate the simulation results of IDEA1. Secondly, the performances of IDEA1 are compared to NS-2 in aspects of accuracy, simulation time and power dissipation analysis. Four metrics are used to evaluate the network performance. They are defined as follows.

- *Packet Delivery Rate (PDR)*: It is the ratio of the number of packets successfully received to the number of packets generated by nodes.
- *Average Latency (AL)*: Latency of a packet is the duration from the generation of the last sensor data in the packet to the receipt of this packet by coordinator. *AL* is an average latency of all packets that successfully received by coordinator.

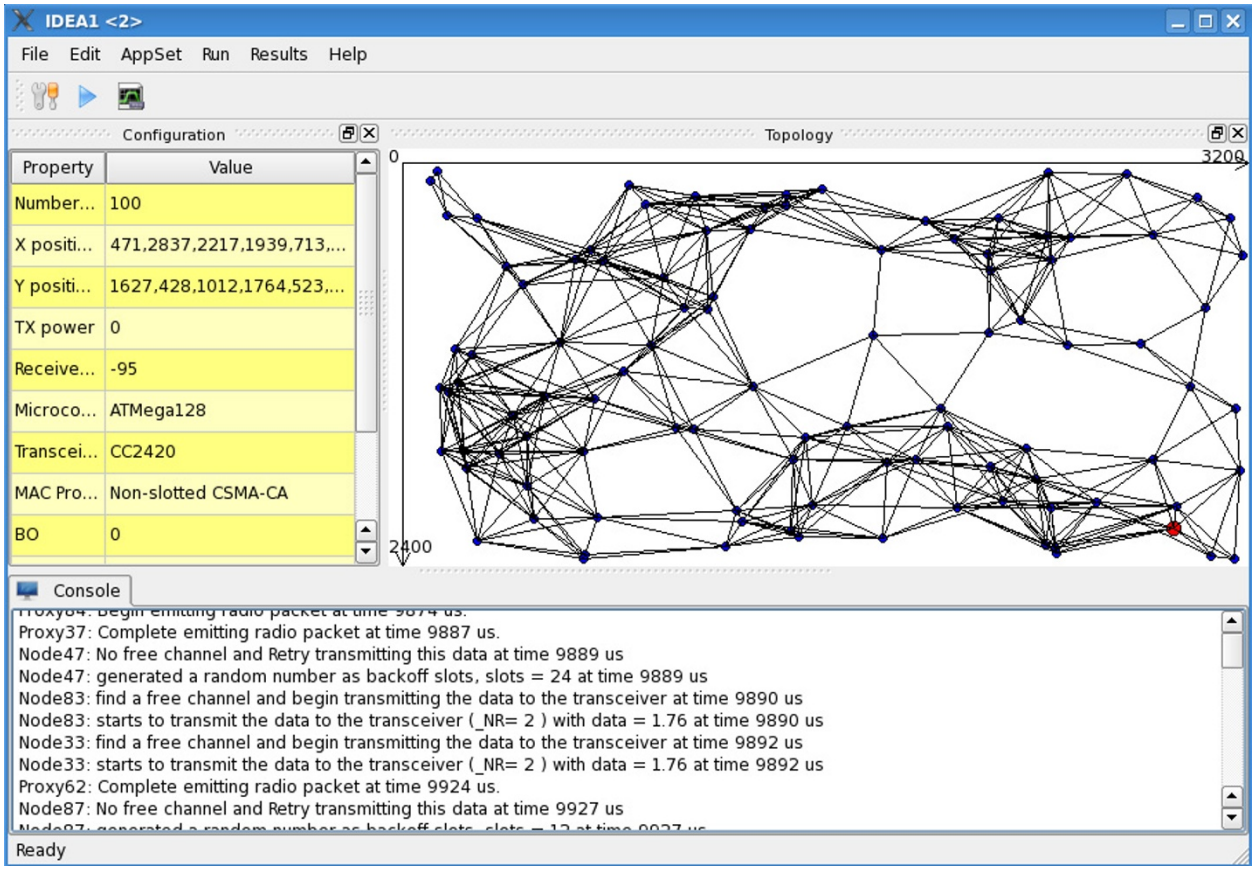

Figure 5 Graphical user interface of IDEA1: a network with 100 nodes is modeled in this example.

- *Energy Consumption per Packet (ECPkt)*: *ECPkt* is the average energy consumed for successfully transmitting one packet.
- *Average Power Consumption (APC)*: *APC* is utilized to measure the average power consumption per node.

4.1 Experimental validation

In this section, the energy model is first calibrated, and then, the simulation results are validated by some experimental measurements of a testbed.

4.1.1 Calibration of energy model

To calibrate the energy model, the current consumptions of every operation mode of hardware components are measured. Our measurement setup is illustrated in Figure 6.

One resistor of 1 Ω was placed series with the power supply of a node (named as node0) in order to measure the current consumption of node0. An instrumentation amplifier [28] with a gain of 76 is utilized to amplify the voltage across the resistor. A Tektronix MSO2012 mixed signal oscilloscope [29] 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 multimeter that can capture extremely small current.

A set of micro-benchmarks have been developed to isolate the hardware consumption of microcontroller and transceiver in order to obtain the current consumptions of each operation mode. The current consumptions of N@L motes is listed in Table 1. As shown in this table, 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.

4.1.2 Validation of simulation results

To validate the simulation results of IDEA1, a testbed of star topology is established. It consists of eight N@L nodes and one coordinator. Nodes send sensor data (an integer of one byte) to the coordinator periodically by using the unslotted CSMA-CA algorithm. The parameters of this algorithm (e.g., *macMinBE, macMaxCSMABackoffs* and *macMaxFrameRetries*) are set as the default values defined in IEEE 802.15.4 standard. The TX power of transceiver is set to 0 dBm. The nodes go to SLEEP mode after the transmission finishes. They are

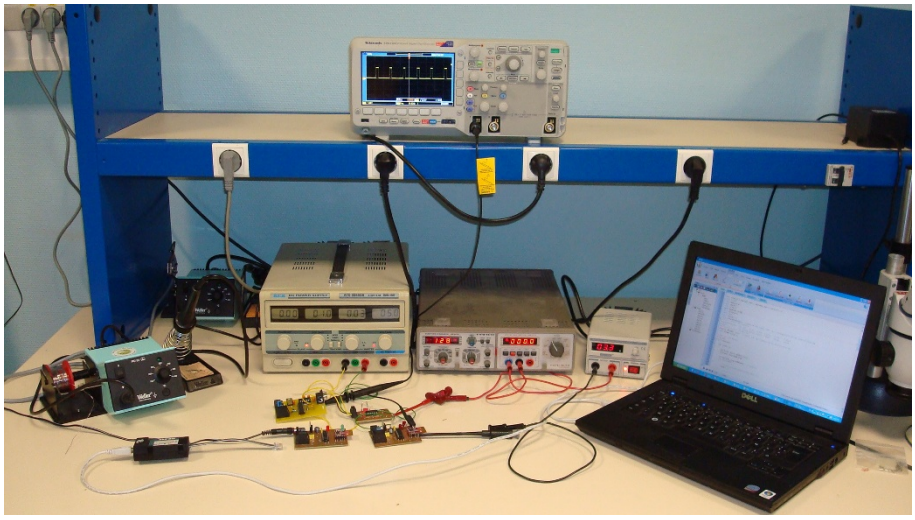

Figure 6 Hardware measurement configuration.

woken up by a built-in timer. It is clocked by an external oscillator in order 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 1,000 Hz. The *ECPkt* and *APC* are measured by the current consumption trace of node0. *AL* and *PDR* can be observed by the output trace of an I/O pin of coordinator, which is also recorded by the oscilloscope. Once the coordinator receives a packet from node0, it toggles one of its I/O pins. The application performed by the testbed has also been implemented in IDEA1 with the same configuration. The simulation and measurement results are presented in Figure 7, 8, 9 and 10.

The average deviations between IDEA1 simulations and testbed measurements are 5.2, 3.2, 6.5 and 3.4%, respectively. Therefore, the average deviation for the four metrics is 4.6% which can be accepted for general 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 sensor data arrives; thus, the *PDR*s and *AL*s 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 power consumption augments due to the decrease in sleep period. The largest sample rate without transmission overlapping is 31.25 Hz.

When the sample rates are 100 or 1,000 Hz, the latency results of experimental measurements are not available. The sample interval is too short that nodes sometimes cannot finish one transmission before the next sensing operation. The nodes may start a new one immediately after a transmission; thus, for a switch of the I/O pin of coordinator, we cannot determine when the sensor data are received by node0. The available testbed and simulation results show that the *PDR*s decrease and the other three metrics augment due to the increase in collisions and less number of packets successfully received by the coordinator. However, the latency is very small when the sample rate is 1,000. In this case, the system is completely saturated, and many new sensor data are read during one transmission. Because the payload size of the packet frame is one, only the latest read data can be sent for the next transmission; thus, the latency is short.

Table 1 Current consumptions of N@L motes (3

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

4.2 Performance evaluation

In this section, the same application of Section 4.1 is studied by NS-2 and IDEA1. The performances of IDEA1 are evaluated by comparing with NS-2. The NS-2 model used in this paper is based on an existing IEEE 802.15.4 NS-2 model in release 2.34 [16]. The model

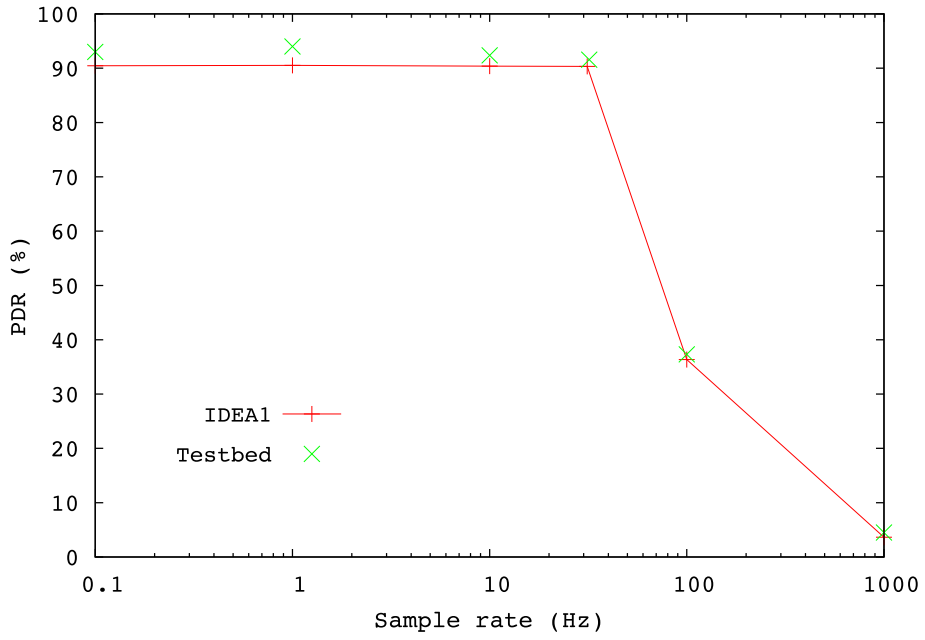

Figure 7 Measured and simulated results of packet delivery rate.

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 [17] have also been added, such as sleep mode and symbol period CCA duration implementation.

NS-2 simulations are written in two languages, C++ and OTcl (Object-oriented Tcl). In general, C++ is used to implement protocol and OTcl is for simulation configuration. Therefore, once the network system implementation is complied as an executable file, many simulations of various configurations of system parameters can be executed

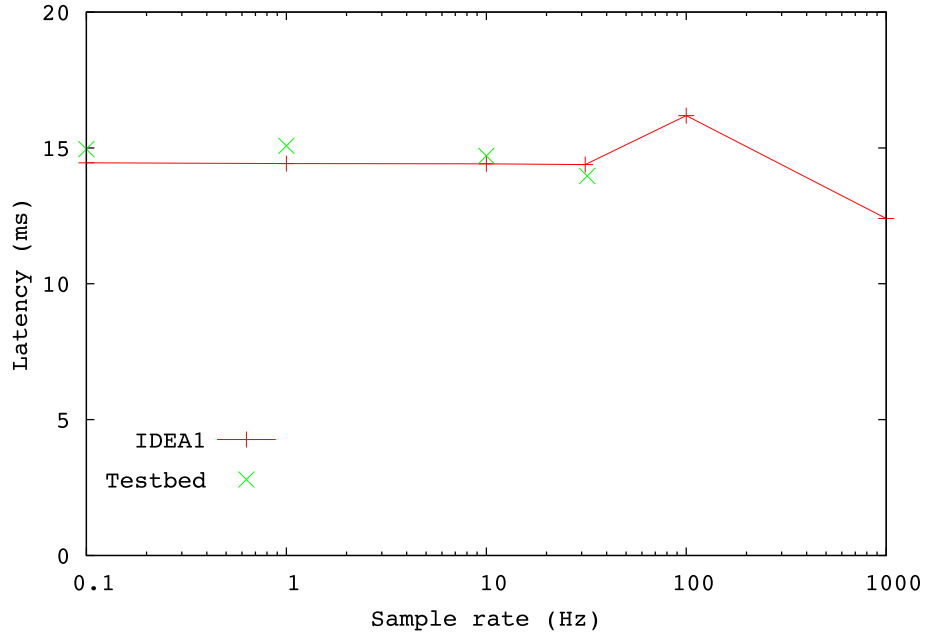

Figure 8 Measured and simulated results of average latency.

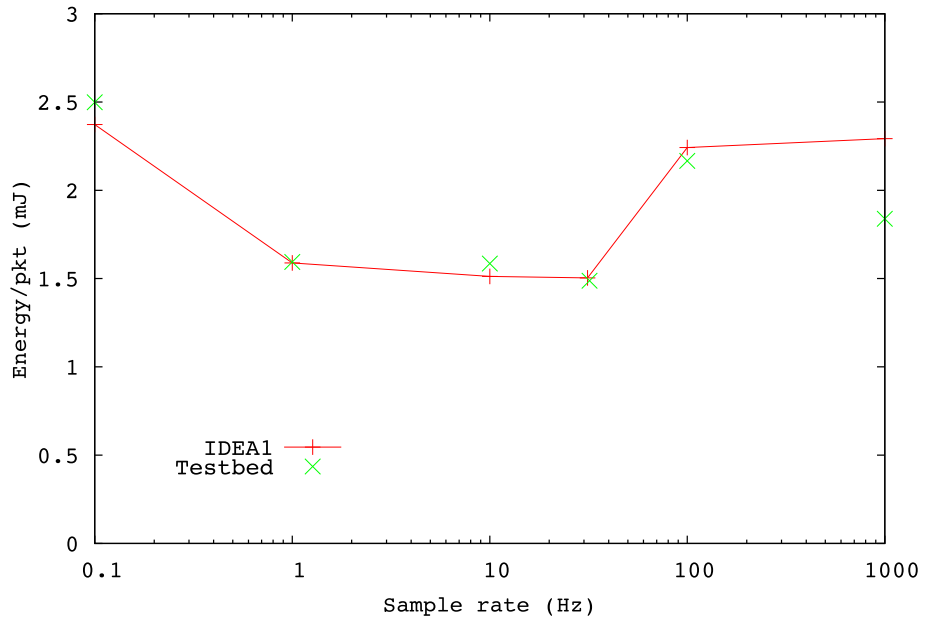

Figure 9 Measured and simulated results of energy consumption per packet.

by just modifying the OTcl script. However, this makes NS-2 hard to learn for beginners. IDEA1 just uses one programming language, C++. All the system parameters are defined in an XML and read by the executable simulation code without recompilation.

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 4.1. The nodes use slotted IEEE 802.15.4 CSMA-CA algorithm to access channel. 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

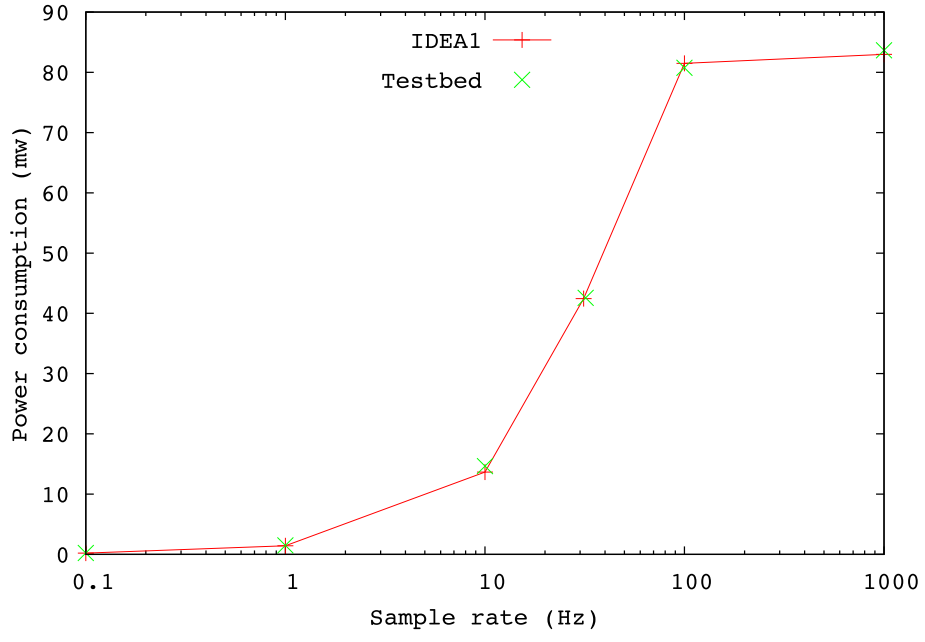

Figure 10 Measured and simulated results of average power consumption.

default in the IEEE 802.15.4 standard. Each simulation includes 10,000 samples, for example, when sample rate is 0.1, the application lasts 27.8 h. Each case is simulated 100 times with different seeds for the generators of random backoff slot numbers.

4.2.1 Simulation results of IDEA1 and NS-2

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 simulation, all the timing parameters about the hardware operations are set to 0, and thus, they do not consume any time or energy. The simulation results are presented in Figure 11, 12, 13, and 14.

In this section, the phenomena illustrated in Figure 11, 12, 13 and 14 are briefly explained. In the next section, the deviation of the simulation results between IDEA1 and NS-2 will be explained. As the sample rate increases from 0.1 to 1,000, the system goes through 3 different stages, i.e., lightly loaded, transition to saturation and saturated.

When 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 *PDR*s and *AL*s remain stable. The *ECPkt*s 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. The *AL* of a bigger *BO* is larger than that of a smaller *BO*. If some nodes cannot transmit their sensor data in one active portion of a superframe, they have to wait until the next superframe to resume their transmissions. A bigger *BO* causes a longer inactive portion.

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 in collisions. The *PDR*s 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. During the period of transition to saturation, the *AL* increases. Some nodes cannot complete their transmissions before the next sample interval begins, and the last one or two nodes have to continue their old transmissions in the new sample interval. Due to more fierce competition of channel usages with other nodes, these transmissions are longer than the cases with a smaller sample rate and *AL* will increase compared with the lightly loaded stage. The smallest energy consumption occurs at the beginning of the transition to saturation. 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

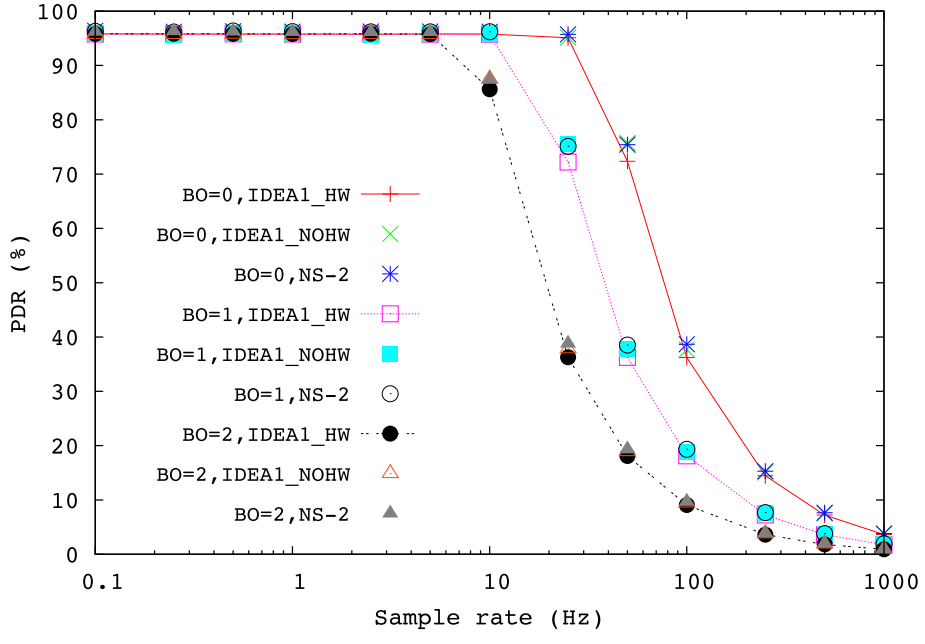

Figure 11 Packet delivery rate simulation results of NS-2 and IDEA1.

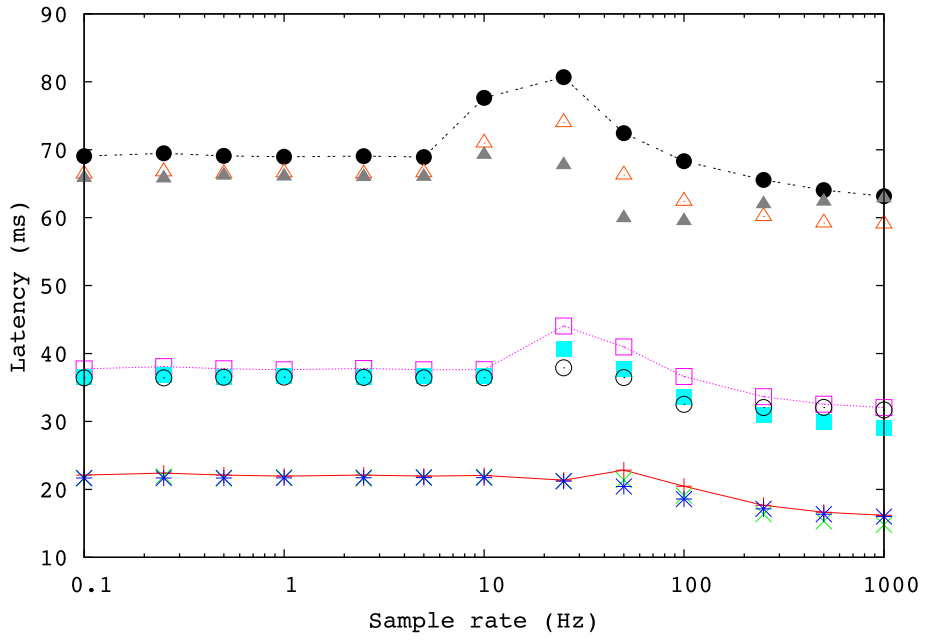

Figure 12 Average latency simulation results of NS-2 and IDEA1.

sample rate continues to increase, the energy consumption augments due to the increase in collisions.

When the system becomes completely saturated, nodes always have several pending data need to be sent. If a node read two new sensor data during one transmission, only the last sensor data will be sent and the first one will be discarded. The duration for the last sensor data stayed in the buffer is smaller; therefore, *AL* decreases. The *ECPkt*s remain constant for the same *BO* because the number of successfully transmitted packets per superframe is almost same. In addition, for a fixed sample rate, since one superframe includes a longer inactive portion if *BO* is larger, its *ECPkt* is bigger.

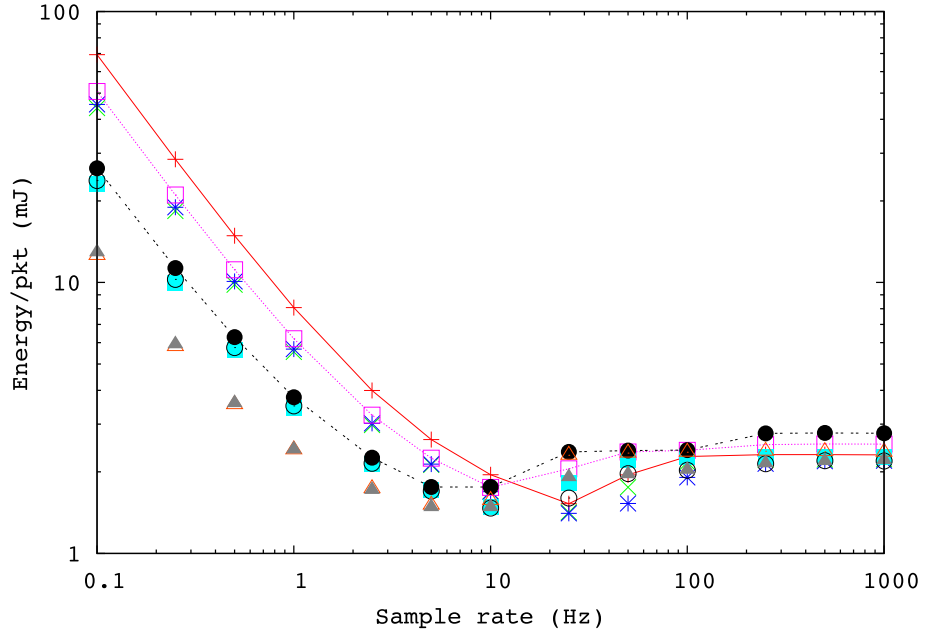

Figure 13 Energy consumption per packet simulation results of NS-2 and IDEA1.

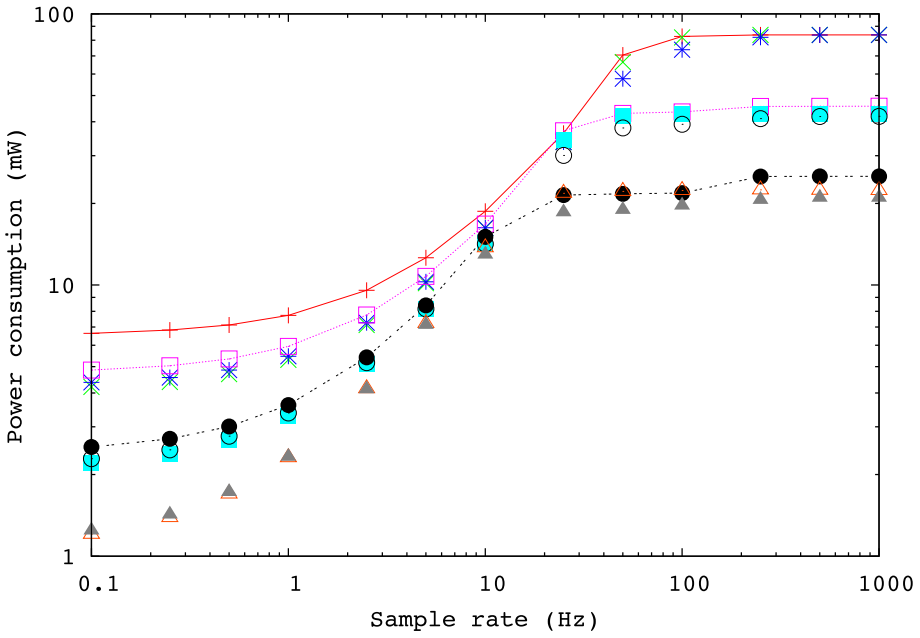

Figure 14 Average power consumption simulation results of NS-2 and IDEA1.

As the sample rate increases from 0.1 to 1,000, the *APC*s augments, because more sensor data need to be sent per unit time. For a same sample rate, the power consumption of a smaller *BO* is larger than a bigger *BO*, since a smaller *BO* means more number of beacon receiving and shorter inactive portion of a superframe.

4.2.2 Result deviations between IDEA1 and NS-2

The deviations of the simulation results between IDEA1_HW and NS-2 of the four metrics are 2.7, 8.9, 49.3 and 45.8%, respectively. The ones between IDEA1_NOHW and NS-2 are 1.0, 2.6, 8.3 and 7.2%. Therefore, the average deviation between IDEA1_HW and NS-2 is 26.7%, and the one between IDEA1_NOHW and NS-2 is 4.8%. The former is bigger since more detailed information of HW/SW operations has been considered. Especially when the sample rate is low, the deviations of *ECPkt* and *APC* results between IDEA1 and NS-2 are very large, because the SPI communications of microcontroller and transceiver account for a very great proportion of the power consumptions. For example, the SPI communication takes 42.4% of the power consumption of microcontroller when sample rate is 0.1. The simulation results proved that IDEA1 provides more accurate modeling of real WSN system. A more detailed power consumption analysis will be provided in next section.

4.2.3 Detailed analysis of power consumptions

In the IEEE 802.15.4 NS-2 model, sensor node is modeled as a whole module and there are no conceptions of hardware components. However, IDEA1 provides more information about the HW/SW operations. The power consumptions of microcontroller and transceiver in different operating modes are presented in Figure 15, 16.

Figure 15 shows the average power consumption of hardware components in different operation modes. The transceiver consumes much more energy than the microcontroller and the energy consumed in the sleep mode is very little.

Besides the power consumptions of active and sleep operating mode, we can also divide the power consumption of microcontroller into more detailed parts, including sleep, analog to digital conversion, SPI communications between transceiver and microcontroller, as illustrated in Figure 16. The microcontroller spend most of its energy for processing data and executing codes, a part of its energy for SPI communication and a little energy for analog to digital conversion and in sleep mode. The consumption of SPI communication is too big to be ignored, especially when sample rate is small.

4.2.4 Simulation time

For the simulations presented in Section 4.2.1, the speed of IDEA1 is about 2 times faster than NS-2. The simulation time of NS-2 and IDEA1 for the application with *BO* set to 2 is presented in Figure 17.

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 h with a sample rate of 0.1 Hz, the simulation times of IDEA1 and NS-

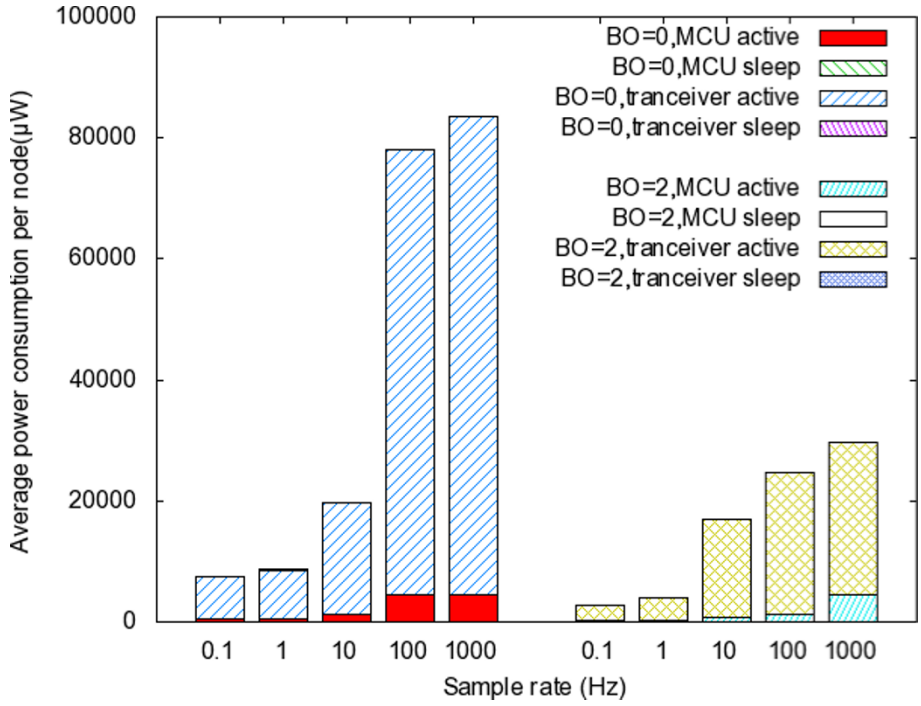

Figure 15 Power consumption decomposition of node.

2 are 7.35 and 24.0 min, 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 while statement of FSM implementation. Instead of checking the states of microcontroller

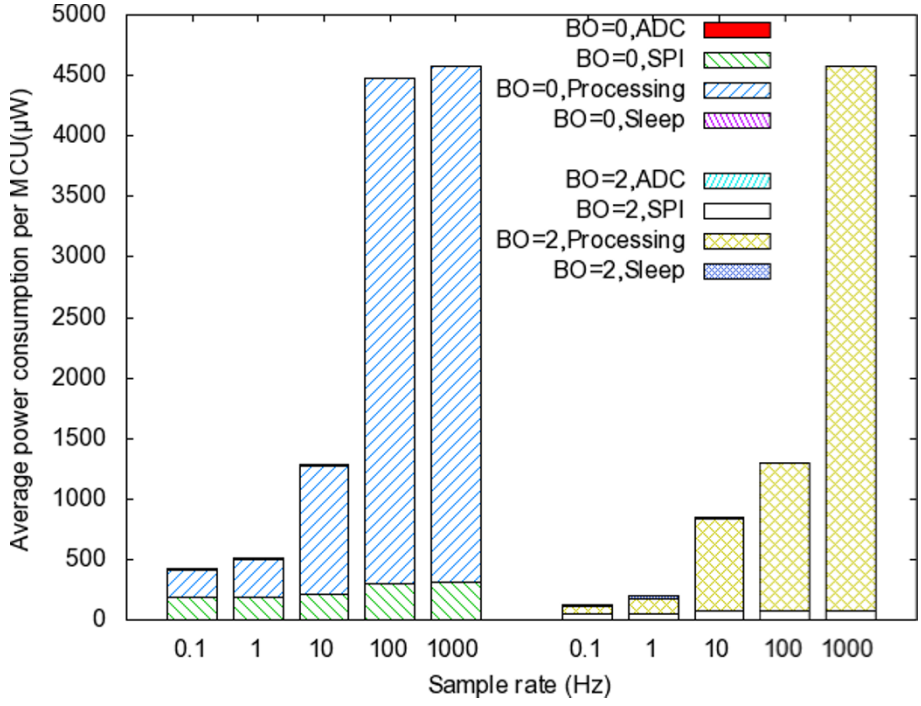

Figure 16 Power consumption decomposition of microcontroller.

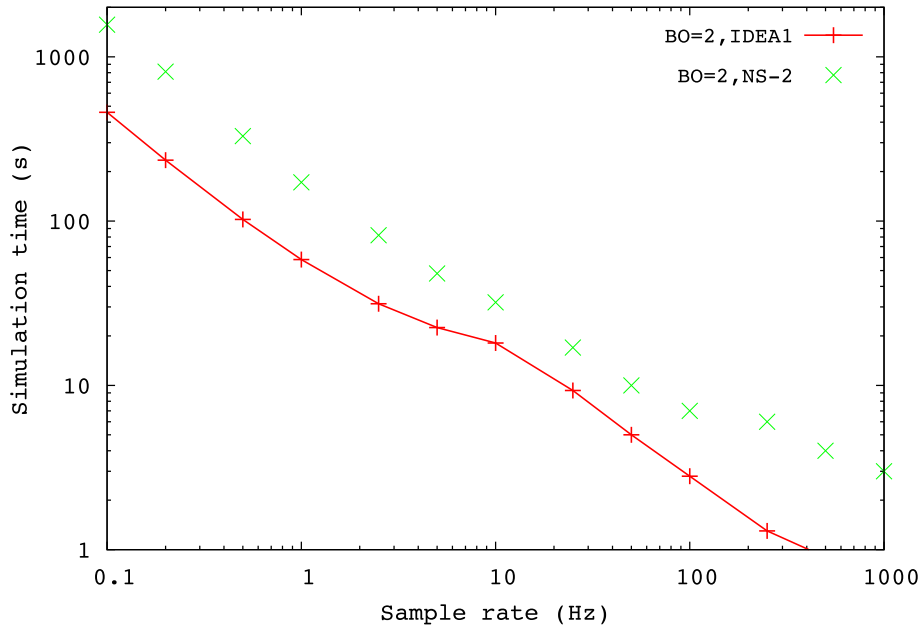

Figure 17 Simulation time of NS-2 and IDEA1.

and transceiver every simulation cycle, their FSMs are woken up only when an interesting event occurs, which can reduce the simulation time significantly.

5 A case study

In previous sections, the simulation accuracy of IDEA1 has been validated by testbed measurements and its performance has been evaluated by comparing with NS-2. Now, IDEA1 is ready to be used in real projects. In this section, an industrial application of WSN in vibration control is studied.

5.1 Industrial application

In Mechanic@Lyon (M@L) project, designers want to identify and integrate new intelligent control technologies in vehicle systems. A wireless sensor and actuator network is deployed on a vehicle to measure and control vibrations, as shown in Figure 18.

The sensor network is composed of several nodes and a coordinator. The nodes measure periodically the vibration of their given positions by a piezoelectric sensor and transmit the data to a coordinator which collects the sensor data of all nodes. The coordinator is connected to a host that analyzes the collected data and implements control algorithms by an actuator network. The main challenges of designing this sensor network are the high sample rate and real-time requirements. The node should read the sensor data with 1 kHz sample rate and send the data to the coordinator within a short latency. At the early stage, some preliminary designs based on the existing hardware platforms and network protocols need to be evaluated.

At first, the eight nodes deployed on the vehicle roof are modeled. The nodes and coordinator form a star topology. All nodes can communicate with the coordinator directly. The nodes store sensor data temporarily in a data buffer and send the sensor data to the coordinator if the data in buffer is more than a certain value (the payload field size of the protocol-defined packet). A sample occupies one byte. *Payload* presents the number of samples in a packet. Since the sample rate is constant, a small *payload* results in more packets to be sent, causing more collisions and thus lower *PDR*. In contrast, a large *payload* leads a longer time for transmitting a packet, which increases the possibilities of channel access failures and causes lower *PDR* too. The best *PDR*, hence, occurs in the case with a moderate *payload*. However, *payload* should be as short as possible. A smaller *payload* means the first sensor data in the packet need to wait a shorter time to be sent.

The application has been tested on the MICAz and N@L motes, respectively, in IDEA1. The goal is to evaluate whether the IEEE 802.15.4 sensor network based on these two platforms can successfully transmit all the sensor data within a short time. The four metrics presented in Section 4 are used to evaluate the network performance: *PDR*, *AL*, *ECPkt* and *APC*. The three MAC algorithms in IEEE 802.15.4 standard are implemented. Because the maximum number of GTSs

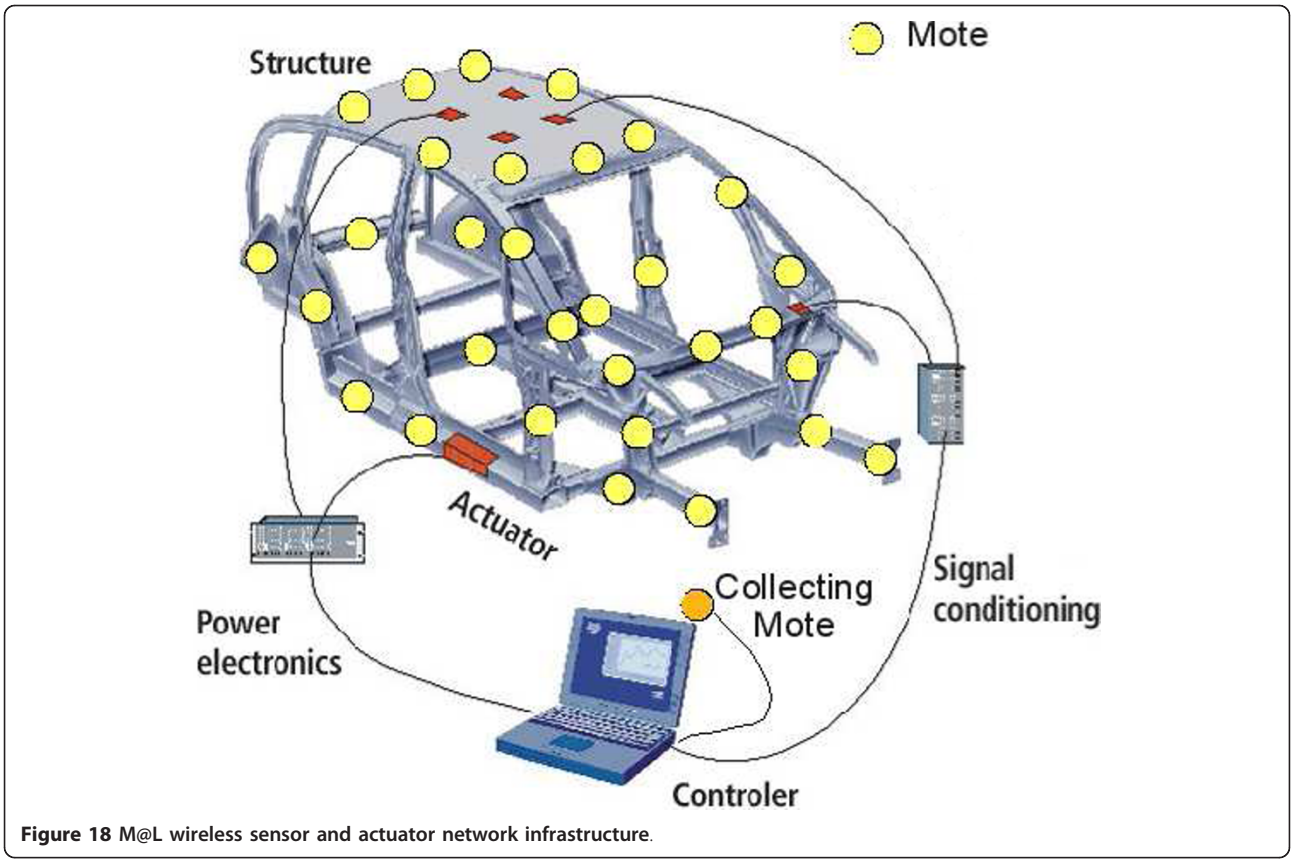

Figure 18 M@L wireless sensor and actuator network infrastructure.

defined in the IEEE 802.15.4 standard is 7, but the current application consists of 8 nodes, a TDMA-based GTS algorithm proposed in [30] is also implemented. It is more suitable for industrial applications which require low packet delivery latency. The contention access period (CAP) is set to 0. The contention free period (CFP) is divided into 8 equal parts, called node slot, which are allocated to nodes off-line. During its slot, the node wakes up if it has data to transmit. Transmissions do not require ACKs since they happen during GTSs without contention. This algorithm can be implemented easily by software on the MICAz and N@L platforms.

5.2 Simulation results

For each algorithm, many cases with different configurations of parameters (e.g., *payload*, superframe length, *macMaxCSMABackoffs*, and *macMaxFrameRetries*) have been simulated. Here, only the best result with the highest *PDR* (or lowest *AL* if two or more cases achieve the biggest *PDR*s) is presented. Each case includes 2500 samples and is simulated 100 times with random seeds. The simulation results of MICAz and N@L motes are provided in Table 2.

5.2.1 Comparisons of MAC algorithms

The CSMA-CA algorithms are not appropriate for this industrial application due to the low *PDR*s, which is

Table 2 Simulation results of OF MICAz and N@L motes

Algorithms	Unslotted	CSMA-CA	Slotted	CSMA-CA	IEEE802154	GTS	TDMA-based	GTS
Results	MICAz	N@L	MICAz	N@L	MICAz	N@L	MICAz	N@L
Payload (byte)	30	30	30	30	30	15	10	19
BO	n/a	n/a	1	1	1	0	n/a	n/a
BI(μs)	n/a	n/a	30,720	30,720	30,720	15,360	10,000	19,000
PDR(%)	36.5	54.4	39.7	67.4	97.4	97.4	100	100
AL(μs)	11,583	15,841	22,426	24,250	53,854	42,777	6,953	12,508
ECPkt(μJ/pkt)	3,811	1,924	3,784	1,576	1,283	1,001	425	408
APC(μW)	46,693	35,155	50,397	35,684	41,071	64,630	42,300	21,264
APC of microcontroller(μW)	29,916	4,576	29,916	4,576	29,915	4,448	29,928	4,573
APC of transceiver(μW)	16,777	30,579	20,481	31,108	11,157	60,182	12,371	16,691

caused by the large number of collisions. The sample rate is too high that the system is overloaded.

Due to the constraints of the maximum GTS slots number of the IEEE 802.15.4 standard, the number of nodes in this simulation is set to 7. The original IEEE 802.15.4 GTS algorithm requires a minimum length of CAP period that is 440 symbols (7.04 ms). Because the packet is transmitted after the CAP portion, the *AL* is high. During one superframe of 30.72 ms, 30.72 sensing operations are performed, but only 30 sensor data can be sent in one packet; the *PDR*s cannot therefore be 100%. This algorithm is implemented by software in MICAz and by hardware in N@L. For MICAz mote, after receiving a beacon packet, the microcontroller can set the transceiver to sleep mode until its GTS slot; however, for N@L mote, the transceiver performs automatically and it keeps in active mode during the CAP portion of a superframe. Therefore, the power consumption of this algorithm based on N@L motes is much bigger than that of MICAz motes.

For the TDMA-based GTS algorithm, the *PDR*s can attain 100%, which prove that the TDMA-based GTS algorithm can reliably transmit the sensor data to the coordinator. However, this IEEE 802.15.4 sensor network fails to meet the real-time requirement of this application. Although the average latency of packets can attain 7.0 ms, *Payload* is 10 samples which mean that the first sample data should wait 17 ms to be received by the coordinator. This latency of sensor data is too high to generate a real-time control action.

5.2.2 Comparisons of hardware platforms

The *PDR*s of N@L are bigger than MICAz, because the MAC algorithms are implemented in MRF24J40 by hardware and the sensing operation in PIC16LF88 has limited impact on the communication process.

The *AL*s of N@L are larger than MICAz, since the SPI communication between PIC16LF88 and MRF24J40 is slower than that between ATMEL ATMega128 and TI CC2420. In order to transmit one packet of several bytes from PIC16LF88 to MRF24J40, the address needs to be sent before each byte. However, ATMega128 only has to transmit one address for a whole packet.

Microchip PIC16LF88 is a power-efficient microcontroller. With an extra low-power-consumption microcontroller, the *APC* s of N@L is smaller than MICAz, although the power consumption of MRF24J40 is much higher than CC2420.

6 Conclusion and future work

This paper presented IDEA1, a 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 COTS hardware platforms have been modeled and the IEEE 802.15.4 standard has been implemented. Energy models based on real experimental measurements have also been developed. The average deviation between the IDEA1 simulations and the experimental measurements is 4.6%. IDEA1 can provide more detailed modeling of sensor network than NS-2 but with less simulation time. The simulation results proved that the hardware and software operations of SPI communication cannot be ignored. By a case study of industrial application, the usability and design process of IDEA1 in real development of WSN systems have been demonstrated. IDEA1 can help the system designers to evaluate some primary designs with low timing and financial cost.

In the future, firstly, we are planning to provide IDEA1 as an open-source tool. To enhance its capability of modeling the real systems, some typical sensor chips will be modeled by SystemC and the interaction between sensors and environment will be studied. More communication protocols, especially high-data-rate protocols, will be implemented.

Competing interests
The authors declare that they have no competing interests.

Received: 15 May 2011 Accepted: 27 October 2011
Published: 27 October 2011

References

1. GP Halkes, KG Langendoen, Experimental evaluation of simulation abstractions for wireless sensor network mac protocols. EURASIP J Wirel Commun Netw. **2010**, 24:1–24:2 (2010)
2. G Sachdeva, R Dömer, P Chou, System modeling: a case study on a wireless sensor network. University of California, Irvine, Tech. Rep. CECS-TR-05-12 (2005)
3. *IEEE Std 1666–2005 IEEE Standard SystemC Language Reference Manual*, (Institute of Electrical and Electronics Engineers, 2006)
4. F Fummi, D Quaglia, F Stefanni, A systemcbased framework for modeling and simulation of networked embedded systems. in *Proceedings of the Forum on Specification and Design Languages* 49–54 (2008)
5. K Virk, K Hansen, J Madsen, System-level modeling of wireless integrated sensor networks. in *Proceedings of the International Symposium on System-on-Chip* 179–182 (2005)
6. J Hiner, A Shenoy, R Lysecky, S Lysecky, AG Ross, Transaction-level modeling for sensor networks using systemc, in *Proceedings of the 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, ser. SUTC '10, Washington, DC, USA: IEEE Computer Society, pp. 197–204 (2010)
7. M Damm, J Moreno, J Haase, C Grimm, Using transaction level modeling techniques for wireless sensor network simulation. in *Proceedings of the Conference on Design, Automation and Test in Europe, ser. DATE '10* 1047–1052 (2010)
8. S Kurkowski, T Camp, M Colagrosso, Manet simulation studies: the incredibles. SIGMOBILE Mob Comput Commun Rev. **9**, 50–61 (2005). doi:10.1145/1096166.1096174
9. *ieee Standard for Information technology-Telecommunications and information exchange between systems- Local and metropolitan area networks-Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, (Institute of Electrical and Electronics Engineers, 2006)
10. P Baronti, P Pillai, VW Chook, S Chessa, A Gotta, YF Hu, Wireless sensor networks: a survey on the state of the art and the 802.15.4 and zigbee

standards. Comput Commun. **30**(7), 1655–1695 (2007). doi:10.1016/j.comcom.2006.12.020
11. L Cai, D Gajski, Transaction level modeling: an overview, in *CODES+ISSS '03: Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, ACM, New York, NY, USA, pp. 19–24 (2003)
12. W Du, D Navarro, F Mieyeville, F Gaffiot, Towards a taxonomy of simulation tools for wireless sensor networks. in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, ser. SIMUTools '10* 52:1–52:7 (2010)
13. K Fall, K Varadhan, *The ns Manual (formerly ns Notes and Documentation)* http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf (Jan. 2009). [Online]
14. A Varga, The omnet++ discrete event simulation system. in *Proceedings of the European Simulation Multiconference (ESM'2001)* (June 2001)
15. S Park, A Savvides, MB Srivastava, Sensorsim: a simulation framework for sensor networks, in *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ser. MSWIM '00*, ACM, New York, NY, USA, pp. 104–111 (2000)
16. J Zheng, MJ Lee, Will IEEE 802.15.4 make ubiquitous networking a reality?: A discussion on a potential low power, low bit rate standard. IEEE Commun. Mag. **42**(6), 140–146 (2004)
17. I Ramachandran, AK Das, S Roy, Analysis of the contention access period of ieee 802.15.4 mac. ACM Trans Sen Netw. **3**, 1–29 (March 2007). doi:10.1145/1210669.1210670
18. V Naoumov, T Gross, Simulation of large ad hoc networks, in *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, ser. MSWIM '03*, ACM, New York, NY, USA, pp. 50–57 (2003)
19. F Xia, A Vinel, R Gao, L Wang, T Qiu, Evaluating ieee 802.15.4 for cyber-physical systems. EURASIP J Wirel Commun Netw 14 (2011). Article ID 596397
20. J Glaser, D Weber, SA Madani, S Mahlknecht, Power aware simulation framework for wireless sensor networks and nodes. EURASIP J Embedded Syst. **2008**, 3:1–3:16 (2008)
21. L Shu, C Wu, Y Zhang, J Chen, L Wang, M Hauswirth, Nettopo: beyond simulator and visualizer for wireless sensor networks. SIGBED Rev. **5**, 2:1–2:8 (2008)
22. P Levis, N Lee, M Welsh, D Culler, Tossim: accurate and scalable simulation of entire tinyos applications, in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ser. SenSys '03*, ACM, New York, NY, USA, pp. 126–137 (2003)
23. J Polley, D Blazakis, J McGee, D Rusk, JS Baras, ATEMU: a fine-grained sensor network simulator. in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004* 145–152 (October 2004)
24. BL Titzer, DK Lee, J Palsberg, Avrora: scalable sensor network simulation with precise timing, in *Proceedings of the 4th international symposium on Information processing in sensor networks, ser. IPSN '05*, IEEE Press, Piscataway, NJ, USA (2005)
25. M Kuorilehto, M Hännikäinen, TD Hämäläinen, Rapid design and evaluation framework for wireless sensor networks. Ad Hoc Netw. **6**, 909–935 (2008). doi:10.1016/j.adhoc.2007.08.003
26. Qt - A cross-platform application and UI framework http://qt.nokia.com/. [Online]
27. Picmicro mid-range mcu family reference manual http://ww1.microchip.com/downloads/en/devicedoc/33023a.pdf. [Online]
28. LT1014 amplifier, (Linear Technology) http://cds.linear.com/docs/Datasheet/10134fd.pdf. [Online]
29. Tektronix MSO2012 mixed signal oscilloscope, Tektronix http://www.tek.com/. [Online]
30. F Chen, T Talanis, R German, F Dressler, Realtime enabled IEEE 802.15.4 sensor networks in industrial automation. in *2009 IEEE International Symposium on Industrial Embedded Systems. IEEE* 136–139 (July 2009)

doi:10.1186/1687-1499-2011-143
Cite this article as: Du *et al.*: IDEA1: A validated SystemC-based system-level design and simulation environment for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:143.