Chapter 9

Power consumption Assessment in Wireless Sensor Networks

Antonio Moschitta and Igor Neri

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/57201

1. Introduction

Wireless Sensor Networks (WSNs) are an emerging technology with a wide range of potential applications. A large number of nodes, with sensing and wireless communications capabilities, deployed in an area of interest, build a WSN. Thanks to the advances in MEMS (Micro Electronics Mechanical Systems) it is nowadays possible to realize small and cheap devices, capable of wireless communication. WSNs differ from other wireless technologies because of a set of specific requirements and characteristic features, including for instance node density, energy requirements, and computing capabilities. The Institute of Electrical and Electronics Engineers (IEEE) classify network technologies by such characteristics. Usually the WSNs are limited to 1Mbps of data rate and 1km of wireless coverage. The actual limit of such quantities depends on the adopted technologies and constraints introduced by specific applications. A set of WSNs specifications, dealing with both network operation and node architectures is described in the IEEE Standard 802.15 and 1451 family. [1-2].

An additional parameter is the WSN operational life, which strongly depends on the balance between power consumption and energy storage. In particular, WSNs are characterized by limited power storage, with possible mitigation coming from power harvesting techniques. Nevertheless, energy efficiency is a critical issue, to be pursued both at node and network level. Typical assumptions include considering the radio interface as the main contributor to power consumption. As a consequence, great attention has been given in the literature to protocol optimization, aimed for instance at minimizing the amount of data transmissions throughout the network, and the maximization of node low-power residence time [3-5]. However, designing a sustainable WSN, relying on power harvesting techniques, would require a deeper and careful modeling, due to the limited and non-steady power supply achievable through harvesting techniques, establishing for instance the maximum consented duty cycle for each node. Such a scenario may require modeling and accurately measuring power consumption associated to the activation of other node functional blocks, in addition to the well-known RF interface. To this aim, both simulation techniques and measurement procedures can be found in the literature. Simulation techniques are available both to describe the network behavior and the node behavior, the latter being based on code profiling techniques and on the description of the node as a finite state machine [6]. Measurement procedures are described as well, typically relying on current measurements at the power supply input, assuming a constant supply voltage [7]. This kind of measurement needs to satisfy conflicting constraints, since it requires to accurately measure short phenomena occurring at a low rate. Moreover, in a distributed context, timing information should be provided, since providing spatial-temporal coordinates to energy consumption measurement may help characterizing the network activity and its operational life.

Following such ideas, the rest of the chapter is organized in three main sections. In section 2, the main features of a WSN are recalled, describing the node architecture and the most popular network topologies, protocols and reference standards. In section 3 energy awareness problems are highlighted, while in section 4 the main techniques for assessing node and network power consumption are recalled, considering both simulators tools and measurement procedures. In the final section, a case study is presented, demonstrating some of the presented approaches.

2. Wireless sensor node and network architecture

2.1. Sensor node structure

Several units compose each node of the WSN, as represented in Figure 1. The core of the wireless sensor node is the processing unit, usually a microprocessor with a limited amount of memory. The processing unit is connected to the sensors via one or more Analog to Digital Converters (ADCs). The sensors and the ADCs form the sensing unit. The data received by the sensing unit are processed and eventually transmitted by the transceiver unit. The transceiver unit is usually capable of bidirectional communications; nevertheless specific applications may require only transmission (TX) or reception (RX) capabilities. Specific nodes may integrate a location finding system that helps the node to discover its position, relative to its neighbors or global. This unit is often embedded on the transceiver module and requires the use of specific algorithms by the processing unit, depending on the adopted localization techniques [8-9].

The power unit and the power generator are a key element in the sensor structure. The power unit is responsible to provide the electrical power needed by the other units in the system. Smart power units are also capable to provide information on the residual available energy, in order to apply energy aware decisions and consent the processing unit to complete the task at hand. Since the power generator usually consists of batteries, such devices have limited amount of energy available, thereby limiting the lifetime of the node. In recent year there has been a big effort in finding alternative solution to power such nodes using the energy available on the node environment with good results [10-12].

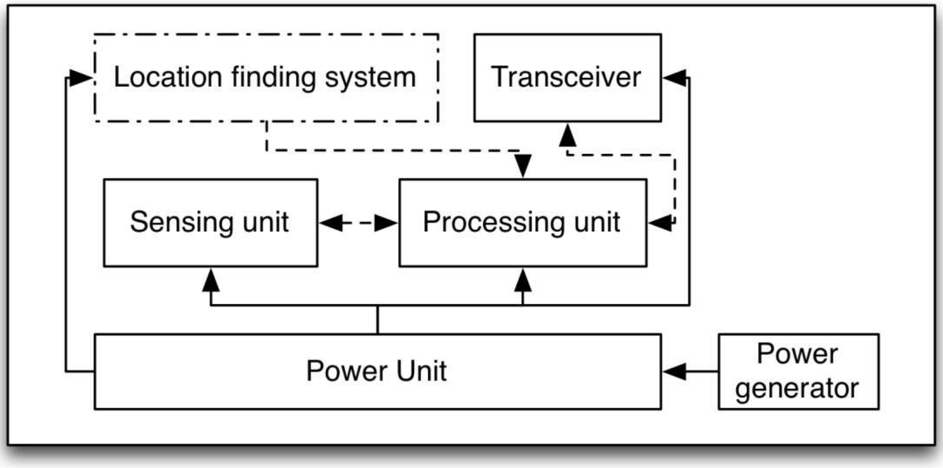


Figure 1. Sensor node architecture

While several off-the-shelf WSN platforms are available on the market, no one of them is considered as a standard de facto. Each research group or industry tends to realize its own platform depending on the objective. This is mostly due to the high cost of existing solutions compared to the costs of the components. Table 1 shows a list of the most popular wireless sensor nodes with their characteristics in terms of computational capacity and radio specifications.

Moreover, it is well known that one of the major constraints in WSN nodes is the low computational capability [13]. Conventional operating systems run on 32/64-bit microprocessors with hundreds or thousands MHz and several MB or GB of memory. For this reason in wireless nodes the applications are specifically designed for the hardware or rely on tailored version of operating systems (OSs), specifically designed for WSN.

Node	CPU	Radio
MICA2	ATmega128L - 16bit 8MHz	868/916 MHz
Telos	TI MSP430 - 16bit 8MHz	2.4GHz IEEE 802.15.4
MICAz	ATmega128L - 16bit 8MHz	2.4 GHz IEEE 802.15.4
ez430-RF2500	TI MSP430 - 16bit 16MHz	2.4 GHz SimpliciTI
WSN430	TI MSP430 - 16bit 8MHz	ISM band (315 to 915 MHz)
VirtualSense	TI MSP430 - 16-bit 25MHz	2.4 GHz IEEE 802.15.4
Pinoccio	ATmega128RFA1 - 8bit 16MHz	2.4 GHz IEEE 802.15.4

Table 1. List of some popular wireless sensors nodes within their characteristics in terms of computational capacity and radio specifications.

Since WSNs can be used to monitor mission critical systems, a Real-Time Operating System (RTOS) is often required. However, only few of the most adopted OSs support RT-applications. According to Farooq and Kunz in [14] the most popular OSs in WSNs are: TinyOS, Contiki, MANTIS, Nano-RK and LiteOS. It is worthy to say that the adoption of an OS increases the power consumption of the node, introducing an overhead due to the management of the process scheduling; however this overhead in power consumption may be compensated by an increase in flexibility in application development. Since different application scenarios call for different tradeoffs, a case-by-case evaluation is required.

2.2. Network architecture

A WSN is usually composed by a large number of nodes deployed in a region of interest. In a typical scenario the region of interest is often a harsh environment, and the nodes are randomly deployed. The sensed data are transmitted through the nodes up to special entities called sinks. The sinks are nodes with two or more network interfaces that act as gateway between the WSN and the user network (e.g. a LAN, or the Internet). The sink usually collects and processes the data from the network sending only relevant information to the user. It also receives commands from the user to be executed on the internal network. A sensor node can communicate directly to the sink (single-hop) or use a multi-hop communication passing the information to its neighbor. Single-hop communication leads to long distance transmission, resulting in high-energy consumption. Using multi-hop communication it is possible to reduce the transmission distance increasing the network lifetime. In multi-hop transmission the network architecture plays a major role. Multi-hop network architectures are typically divided in flat or hierarchical as represented in Figure 2.

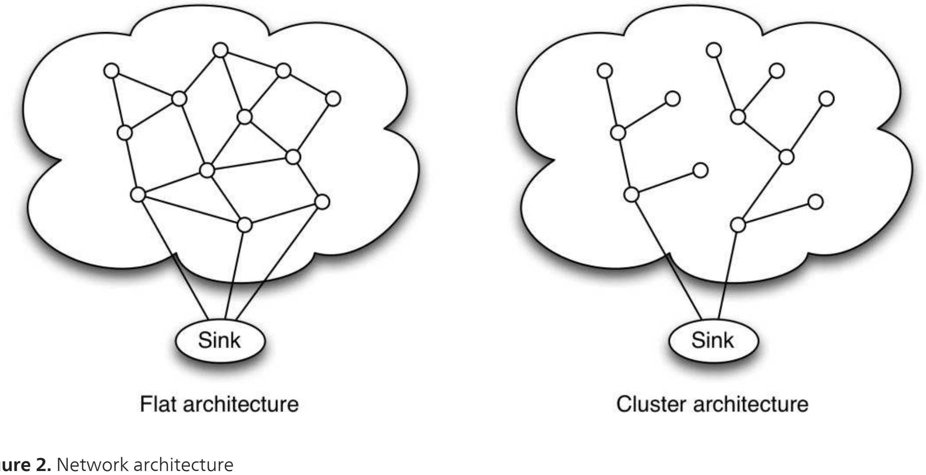


Figure 2. Network architecture

In flat architecture each node plays the same role in sensing and transmitting the information. In hierarchical architectures the nodes are organized into clusters. In each cluster, one or more nodes (head node) are responsible to communicate with other clusters or directly to the sink. The head node may be dynamically selected by various criteria, including its available energy, distance between cluster members and other cluster heads [15], and node homogeneity [16-17].

2.3. Wireless sensor network standards

Interoperability between different products is guaranteed by compliance to standards. In this way different sensors, produced by different manufacturers can communicate and achieve a common objective in the WSN. During the last years different WSN standards have been developed, taking into account different objectives, levels of abstraction and node components. IEEE defines two standard related to WSNs, the IEEE 802.15 regarding the wireless communication interface between nodes and the IEEE 1451 that defines the interface between sensors and actuators. Among the several subgroups of the IEEE 802.15 family the most important, for the WSNs is the IEEE 802.15.4, which specifies the physical layer and media access control (MAC) for low-rate wireless personal area networks (LR-WPANs). The standards support different frequency bands, number of channels and data rates as represented in Table 1.

Band	868 MHz	915 MHz	2.4 GHz
Region	EU, Japan	US	Worldwide
Channels	1	10	16
Data rate	20 kbps	40 kbps	250 kbps

Table 2. Frequency bands, number of channels and data rates for the IEEE 802.15.4 standard.

The preferred frequency band for WSNs is the 2.4 GHz since it is worldwide usable and it has the highest data rate. Moreover the high radio data rate reduces frame transmission time, reducing the microcontroller idle time, resulting in overall energy consumption reduction.

Based on the IEEE 802.15.4 standard several specifications have been developed by different consortia. One of the most famous is ZigBee, a protocol that defines the network and the application layers, built upon the IEEE 802.15.4 physical and MAC layers. Following the ZigBee specification the network layer provides support to tree, star, point-to-point (mesh) network topologies using three different kinds of nodes:

- Network coordinator: it forms the root of the network tree and might bridge to other networks.

- Router: it can act as an intermediate router, passing on data from other devices.

- End device: it cannot relay data from other devices; it is only able to communicate with its parent (router or coordinator).

The application layer provides a framework for distributed application development and communication. On the application layer it is possible to develop up to 240 application objects, which are user defined application modules implementing a ZigBee application. In this way several different applications can reside on a single node, sharing the lower stack of the protocol. Each application object in the network is identified by the network address of the hosting device and the application endpoint number (from 1 to 240).

3. Energy awareness and control of power consumption

The number of existing and prospecting applications has been steadily growing after the development of the WSN paradigm. Regrettably, the energy density of the batteries did not follow the same trend, and the energy harvesting systems can power only a limited class of devices, usually with limited capabilities [10]. For this reason energy consumption modeling and reduction has attracted the interest of both the academic and the industrial worlds. The next sections are devoted to the exploration, modeling, characterization and analysis of the power consumption of a WSN node in relation to specific application. In this section a brief and non-exhaustive review of methods to reduce the power consumption of the nodes is presented.

Due to the limited computational capabilities of the WSN node its load is often limited to trivial computation. The greatest part of energy is spent by the peripherals, especially by the radio module. Thus, a lot of power-saving mechanisms exploit the energy consumption reduction of the node peripherals. In this regard, both passive and active approaches are possible. Passive power conservation mechanisms reduce the energy consumption of a sensor node by turning-off its transceiver interface module when there is no communication activity [18]. Moreover, additional energy savings may also be achieved by optimizing the performance of the processor in an active state changing its operational frequency [19]. In fact, using a processing unit with variable processors speed (VPS), it is possible to decrease its power consumption decreasing the supply voltage and the clock frequency. Exploiting the VPS it is desirable to design a scheduling system, capable to select a suitable supply voltage and relative frequency clock for each task. Dynamic Voltage Scheduling (DVS) is one of these mechanisms able to provide such behavior without degrading the overall performance of the node [19]. Dynamic Power Management (DPM) is another technique to increase the lifetime of a sensor node [20]. DPM acts similarly to DVS, but instead of scaling the clock frequency it can dynamically turn-off the components of the sensor node and wake them up when needed. At microcontroller level this transition of states it is represented by different power mode that shutdown the CPU, memory or additional internal peripherals. It is worth to say that each transition of state takes a certain amount of time and consequent energy consumption as reported in Figure 3. In each power mode, also called low power mode (LMP), different peripherals are incrementally turned off. Each transition from the idle state to a LPM has a fixed cost, indicated in Figure 3 as *b0*, which is usually negligible. However the energy cost for waking up the microcontroller from a low power mode increases with the depth of the low power modes. For this reason it is important to reduce the number of state transitions, conveniently balancing the scheduling mechanism without using aggressive power down strategies.

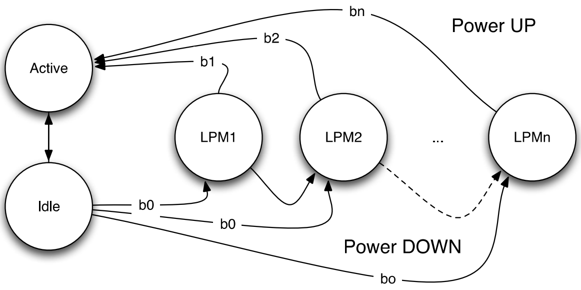


Figure 3. Low Power Modes (LPMs) transitions and costs

Active power conservation mechanisms differ from passive ones in that they achieve a reduction of the energy consumption by avoiding undesired events like collisions, or exploiting energy-aware routing protocols. For instance adjusting the transmission power may help minimizing the probability of occurrence of a collision, an event leading to higher power consumption due to the related detection and retransmission activities. Multiple Access with Collision Avoidance [21] (MACA) and Multiple Access with Collision Avoidance Wireless [22] (MACAW) are two different MAC layer channel access protocols, aimed at avoiding or minimizing the collision rate by using a particular handshake signaling. Conversely, Power Controlled Multiple Access [23] (PCMA) is a MAC protocol that can achieve power-controlled transmission and thus collision avoidance, originally proposed for ad-hoc networks but suitable to WSNs as well.

Operating at PHY level and exploiting the frame filtering technique, it is possible to achieve a substantial reduction in energy consumption. Usually receivers perform the channel clear assessment (CCA) in order to check for incoming packets or avoid collision. The IEEE 802.15.4 standard defines three possible methods to perform the assessment:

- Energy above threshold. If the energy detected is above a fixed threshold the CCA shall report a busy medium.
- Carrier sense only. This method checks for a signal with modulation and spreading characteristics of the IEEE 802.15.4. In this case the signal may be above or below the threshold.
- Carrier sense with energy above the threshold. This is a combination of the previous methods checking both signal characteristics and energy.

Once the CCA reports a busy channel the receiver may start its RX phase to obtain the packet content. It is clear that in a dense network, where a lot of transmissions occur, there are many chances to detect a transmission. In this case most of the packets sent on the network are not intended to the receiver itself but to others receivers, generating unintentional package reception. Each package reception is an energy expensive procedure and for this reason it may be reduced at the minimum avoiding unintentional package. In [24] the authors exploit the characteristic of the Texas Instruments CC2520 RF transceiver of executing specific operations once the packet header is received. In particular, the authors modified the firmware of the RF transceiver in order to trigger an interrupt when the packet header reports as recipient an address different from its own. In this case the RF transceiver sends an interrupt to the MCU that turns off the radio module saving the energy needed to receive the packet payload. The energy consumption of CCA, result of frame dropping, unintentional and intentional package is represented in Figure 4. In subFigure 4.c, a representation is shown of the scenario where an unintentional package is received. In fact after reception the MCU does not perform any task, the opposite of what happens in scenario 4.d, where a payload processing is performed. Figure 4.b represents the case where the RF module triggers an interrupt on the MCU, saving the energy cost for the payload reception.

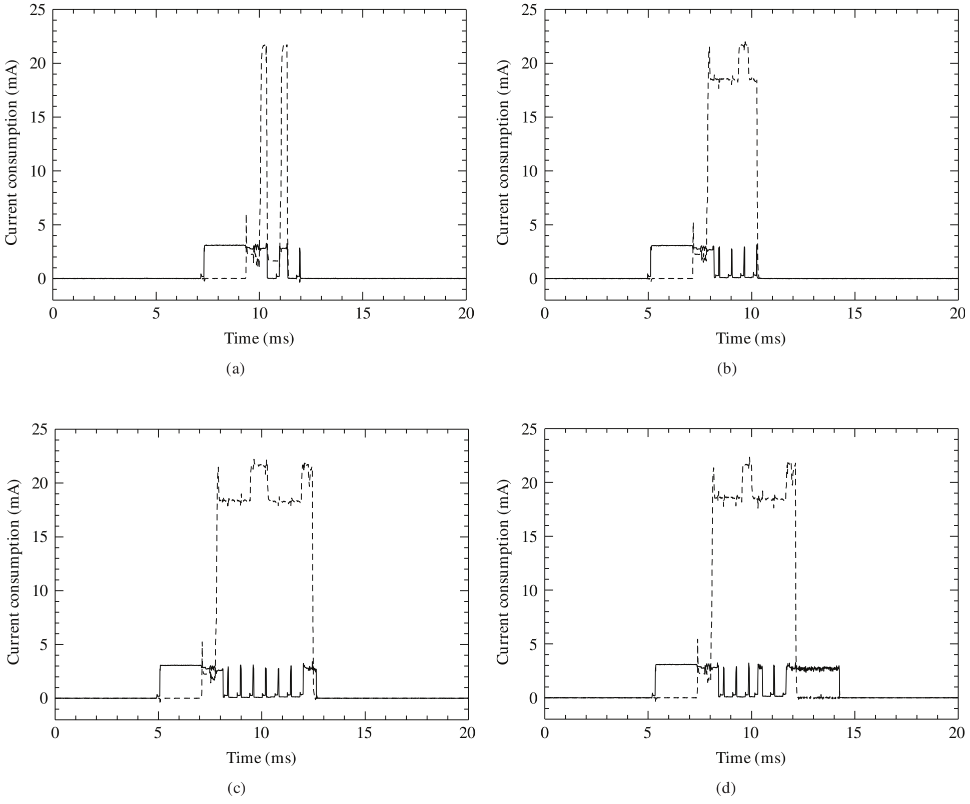


Figure 4. Power consumption of MCU (solid line) and radio transceiver (dotted line) during: (a) channel clear assessment, (b) frame filtering, (c) broadcast reception, and (d) broadcast reception and processing. [24]. Copyright © 2012 Emanuele Lattanzi and Alessandro Bogliolo, distributed under the Creative Commons Attribution License.

At routing level several mechanisms were proposed in last years to increase the lifetime of the WSN, increasing the lifetime of the nodes' batteries [3-5]. Most of the proposed techniques take into account nodes powered by batteries but the routing strategies may change taking in consideration nodes energetically sustained by their environment changing. In this scenario the routing mechanism has to be dynamically selected taking in consideration not the total energy available on the batteries but the energy available for each node at specific time (i.e. the power available). In [25] the authors demonstrate that power-constrained WSNs, as nodes powered by energy harvesting, can be represented as flow networks and that the optimization of the energetic sustainability of the workload can be cast into an instance of maxflow. Starting from this consideration, Lattanzi et al, in [26], propose a non-deterministic routing table that can be actually applied at the sensor nodes in order to achieve the maxflow theoretical optimum. In this case, since the information of the available power is accessible only to the relative node, the nodes have to cooperate to solve the maxflow problem.

4. Modeling and measurement of power consumption in WSNs

In order to ensure the expected lifetime in a WSN it is important to properly define the workflow of the nodes, evaluating and measuring their power consumption. Such evaluation may provide feedback during application design phase, consenting to improve the overall energy efficiency. The power consumption profiling of a node is also an important stage in the deployment of a WSN, since it consents to properly configure the duty-cycle and the number of transmissions as a function of the available energy. There are several methods to estimate the power consumption of a WSN node, including theoretical estimation, direct measurements, and usage simulations tools.

Theoretical estimation relies on an abstraction of the network, including the surrounding environment. However, due to the difficulty of describing the environment, realistic models are not easily realized and evaluated and even simplified models can be very complex, resulting impractical, or not accurate [27].

Direct measurements, relying on physical sensor node, offer the best accuracy on energy consumption estimation and evaluation, and are often used. Due to the complexity of the network sometime measuring the energy consumption of a whole sensor network results a very complex task. Not only measurements should be collected in different places, but WSN state and distributed power consumption measurements may require a common time reference shared by the involved nodes, so that local measurements are properly synchronized. A hybrid framework, envisions single node measurements, to be carried out with an oscilloscope or specific instrumentation under fixed conditions. Then measurement taken on a single node may be projected to the entire WSN only under some specific conditions (e.g. when the WSN nodes are homogeneous ad performs similar task). A wide measurement campaign can be carried out on limited size WSNs using specific systems [28].

Due to the variety of available platforms and environmental constraints, the design, implementation and deployment of a sensor network application are complex tasks. Thus it is often useful to simulate, at various stage of development, one or more components of the networks. Thus, accurate simulators may be a useful tool for the assessment of the WSN performance, especially given knowledge of available energy source, and the achievable duty cycle and operational life. Typical requirements are then accurate simulation of network behavior in response to specific events, accurate simulation of individual node behavior, and time awareness. Moreover, a WSN node typically includes mixed signal processing devices. As anticipated, a well known mixed signal contributor to power consumption is the radio interface, responsible for a large portion of energy consumption. Other significant contributors may be active sensors and A/D converters, whose conversion time, on a microcontroller platform, may be lower than a microsecond. In this case, not only a low level finite state machine should be modeled, but, for calibration purposes, an accurate measurement system is needed, capable of tracking state transitions lasting a few microseconds.

Specific solutions have been proposed in recent years for various wireless systems. For instance, some simulators have been developed for PAN network devices, such as Bluetooth. In [29], a Bluetooth device has been described as a finite state machine, each state being associated to link manager level activities, such as a scan/inquiry operation. Then, average power consumption was measured for each of the identified state transitions, each lasting a few milliseconds, using Digital Multi Meters (DMM). As a result, the average power consumption of a Bluetooth device executing a given application could be predicted with good accuracy [29]. While effective, such approach features a large time granularity that may be suboptimal for WSN applications. In fact, WSN nodes are often arranged in a peer to peer or mesh configuration, where several asynchronous and short events may occur, and featuring various low-power/sleep modes. Moreover, a deep optimization of power consumption may require a simulation tool to profile the energy cost of the internal work of each node. This requires to model events with time constants that may be lower than a microsecond [30].

Thus, other WSN simulators have been recently developed, focused on the simulation of the protocol and MAC level, on processor profiling, on attempting to combine both features [31]. In network focused simulation frameworks, sensor nodes are generally represented using a layered architecture, where each layer is responsible to model specific hardware or software aspect of the node. Moreover, in order to study the energy consumption profile of the node, accurate timing information is needed. Thus, such simulators alone, being oriented to model the network activity and the information flow, lead to a coarse representation of the node states, and are not suitable for accurate energy consumption estimation. Another class of simulators emulates the platform executing the same code of the node. Using this technique it is possible to obtain a fine-grained timing, permitting the simulation of interrupts and low-level peripheral interaction. Such simulators are usually called instruction-level simulators. Due to the strict hardware dependence each simulator is usually capable to emulate only a few platforms, relying on configuration files that describe the peculiar characteristics of a given platform. It should be noted that, in order to obtain an accurate simulation of power consumption, also platform components embedded in the WSN node with the CPU may be significant contributors to power consumption, and should be properly kept into account. An example is provided in section 5, where a case study is discussed.

Finally, simulation tools should be coupled to proper measurement techniques, keeping into account potential and limitation of the available instrumentation. For instance, in [29] accurate measurements have been carried out using DMMs, and measurement uncertainty has been modeled by describing the effect of measuring phenomena of comparable duration with the DMM integration time. More generally, the requirements of a measurement system should include accuracy, and the capability of capturing and measuring phenomena with short duration and a potentially low repetition frequency.

In the next subsections different methods to evaluate by experiment or by simulation the energy consumption of a WSN node are presented.

4.1. Measurement setup for current consumption

Since embedded systems usually operate at constant supply voltage, power consumption measurements can be carried out indirectly, by measuring and monitoring the absorbed current. To this aim, various techniques are available, described in the following.

A very common solution is the series insertion of a small resistance R_1 (≤10Ω) between the power supply and the Device Under Test (DUT), as shown in Figure 5. Then, by measuring the voltage drop ΔV=V_2-V_1 across the resistor, current I can be measured indirectly, using Ohm's law, as

$$\hat{I} = \frac{\Delta V}{R_1} = \frac{V_2 - V_1}{R_1} \tag{1}$$

and the absorbed power can be estimated as $\hat{P} = \hat{I} \cdot V_0$.

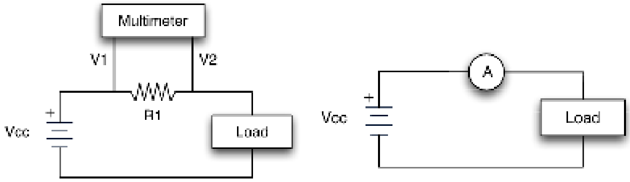


Figure 5. Measurement setup with a shunt resistor or amperemeter

Such measurements, performed during normal operation of the platform, allow monitoring the power consumption when different components of the board are active, and its time dependence. A key role in the reliability of such measurement is played by the voltage measurement system accuracy, the tolerance on the shunt resistor value, the stability of the supply voltage, and the measurement rate, that should be compatible with the analyzed phenomena. Notice that, with such a system, the voltage drop across the resistor reduces the supply voltage powering the DUT, introducing a type B contribution to the measurement uncertainty for large values of R_1 [32]. On the other hand, low values of R_1 lead to reduced values of ΔV, resulting in lower measurement sensitivity. By assuming that the aforementioned offset effect is negligible, and that uncertainties on R_1, V_0, and ΔV are uncorrelated, using the law of propagation of uncertainties the measurement uncertainty $u(I)$ on current I and the measurement uncertainty $u(P)$ on power consumption P are respectively given by [32]

$$u(I) \cong \sqrt{u(\Delta V)^2 \frac{1}{R_1^2} + u(R_1)^2 \left(\frac{\Delta V}{R_1^2} \right)^2} \tag{2}$$

and

$$u(P) \cong \sqrt{u(\Delta V)^2 \left(\frac{V_0}{R_1} \right)^2 + u(R_1)^2 \left(\frac{V_0 \Delta V}{R_1^2} \right)^2 + u(V_0)^2 \left(\frac{\Delta V}{R_1} \right)^2} \tag{3}$$

where $u(\Delta V)$, $u(V_0)$, and $u(R_1)$ are uncertainties on ΔV, V_0, and R_1 respectively.

A similar approach to the resistor method is the direct insertion of an amperemeter, capable of measuring currents ranging from the microampere, the typical absorption of a microcontroller settled in sleep mode, to a few tens of milliampere, corresponding to a full workload (data collection/processing, RF transmissions). Uncertainty contributions may be evaluated using an approach similar to that associated to the resistor method. Notice that, depending on the amperemeter architecture, bandwidth limitations may lead to averaging of the measured current, leading to a loss of information [30].

In order to improve the measurement accuracy, alternative approaches have been suggested. For instance, in [33] a method has been proposed, based on inserting a switched pair of capacitors between the power supply and an ARM7TDMI processor, as shown in Figure 6. By alternatively switching the capacitors C_{S1} and C_{S2} with the microcontroller clock, the processor can be powered by the capacitors. By also keeping into account the effect of the on-chip capacitance, the energy consumption can thus be estimated by measuring over time the voltage drops across both capacitors, and by recalling that the energy stored in a capacitance C with a

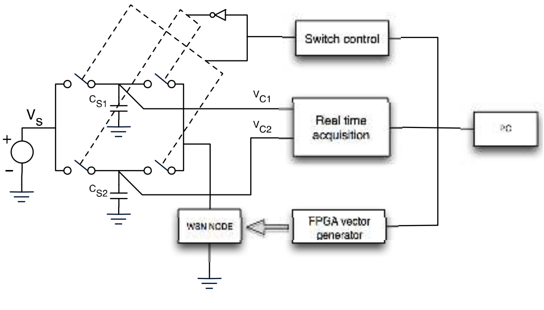


Figure 6. Measurement setup with switch capacitor

voltage drop *V* is given by $CV^2/2$. This method removes the offset uncertainty introduced by the resistor method.

Another method to reduce measurement uncertainty has been proposed in [30]. Here, a current mirror has been designed as shown in Figure 7, whose symmetric topology replicates the current absorbed by the microcontroller. Such replica is then measured, without perturbing the microcontroller power absorption. In this case accuracy is limited by tolerances of the current mirror components, which should be carefully matched in order to guarantee an accurate replication of the current absorbed by the DUT.

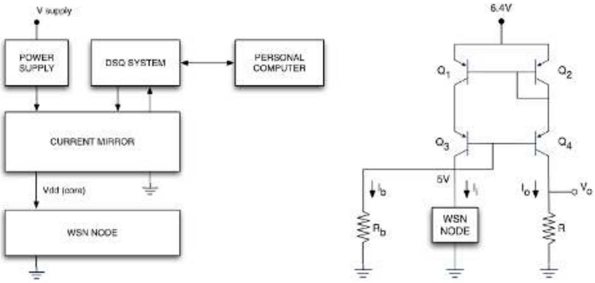


Figure 7. Diagram and circuit of measurement setup with current mirror

It should also be observed that the proposed approaches are suitable for node level measurements, but may be unpractical in large WSNs, especially if deployed in large of harsh environments. In this case specific solutions are used. Hergenröder et al. in [28] presented a distributed energy measurement system called Sensor Node Management Device (SNMD), used in conjunction with the SANDbed testbed [34]. SNMD is a measurement system suitable for different sensor node (e.g. MicaZ, IRIS, SunSPOT), providing energy measurement on individual nodes. The current measurement is based on shunt resistor (i.e. 1 Ohm) approach as described before. The resulting system has a current selectable range up to 500mA on 0-10 Voltage range with 16bits of resolution and a sampling rate up to 500kHz (20kHz without buffering). A key aspect in distributed measurement, using a SNMD device on each WSN node, is the synchrony between the measurements on different nodes. For this reason on SNMD the time is synchronized using the Network Time Protocol (NTP) that provides an accuracy of 10ms. However such accuracy can be too coarse for specific applications, especially in relation to the high sampling frequency of the node. For this reason the authors in [28] proposed the use of offline algorithms to synch and analyze measurements performed on different nodes.

An effective measure of current consumption of a WSN node during operation is presented in Figure 8.

4.2. Modern energy consumption simulation software

As mentioned before, the simulation software depends on the considered node architecture. In this section we present two different instruction level simulation software, developed

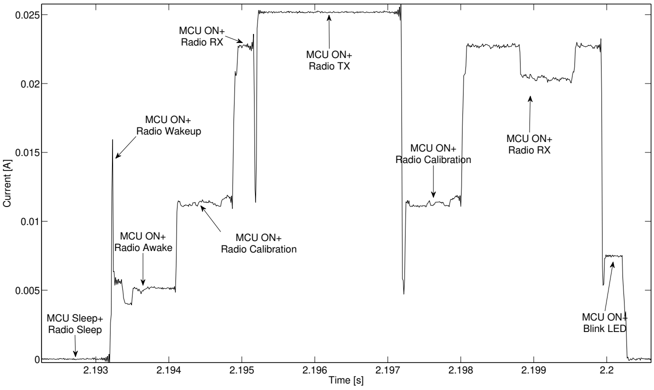


Figure 8. Current consumption measurements for a WSN node during transmission with acknowledgement

respectively for the AVR microcontroller, produced by Atmel, and for the MSP430 microcontroller, produced by Texas Instruments.

The first considered simulator, known as Avrora, is a set of simulation and analysis tools developed the UCLA Compilers Group [35]. In particular, the simulator can handle up to a few thousands nodes, by taking advantage of the processing power of modern computers. Avrora is not only a simulator to test program execution on the node but it also allows online monitoring of the code execution on the WSN, profiling utilities to study the program's behavior, source level debugging, a control flow graph, providing a graphical representation of program's instructions, and an energy analysis tool, capable of analyzing the energy consumption of a specific application.

The simulator has been enhanced by Haas et al., after evaluating the performance of the analysis tools of Avrora, comparing the simulation results with experimental measurements carried out with the SANDbed platform [36]. The test application, run over TinyOS and involving four nodes with fixed routing path. Using the collected data Haas et al. developed and released an enhanced version of Avrora, called Avrora+, improving the calibration of energy model, modeling transition state cost, and taking into account the effect of manufacturing tolerance on the energy consumption.

The experimental verifications showed that the Avrora+ is very accurate, reducing the difference between measurements consumption measurements and simulation results to less than 5%.

The Worldsens simulation Framework is another WSNs simulator that support MSP430 based node [37]. This open source platform, released under the CeCILL and GNU GPL license agreements, includes three simulation tools, often used in conjunction:

- WSim: this is the platform simulator. It performs a full instruction simulation of the node, driven by the microprocessor internal clock.

- WSNet: an event driven wireless network simulator that can be used in conjunction with WSim to simulate a whole sensor network with high accuracy.
- eSimu: a software module that implements platform specific energy consumption models, and provides an estimation of the current absorbed by a node [38].

For estimating the power consumption, WSim and eSimu are usually being jointly used, interacting as shown in Figure 9. In particular, the WSim tool, compiled with eSimu support, receives the binary file that would be executed by a real microprocessor for a given application and provides a trace file, describing state transitions of the node and its peripherals. Notice that, when modeling a radio transmission, WSNet is used as well, in conjunction with WSim. Using the trace file and a calibration file reporting the current absorbed by the node in its various states, the overall current consumption of a node that executes a given task can be estimated and profiled against the execution time. Since the node is powered by a constant voltage source, the power consumption can easily be derived from the current absorption.For an exhaustive overview of existing tools for simulations, modeling and measurements of WSNs refer to [39].

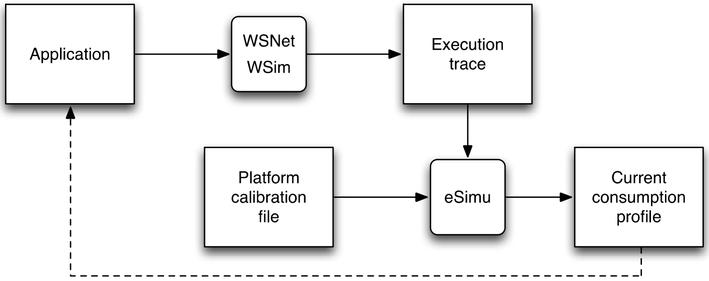


Figure 9. Wsim, WSNet and eSimu simulation process

5. Case study

In this section we present experimental measurements along with simulation results regarding one of the platforms mentioned in section 2: the ez430-RF2500 development kit from Texas Instruments. The considerations discussed remain valid for similar nodes and may be applied to different architectures or cases.

5.1. Platform description

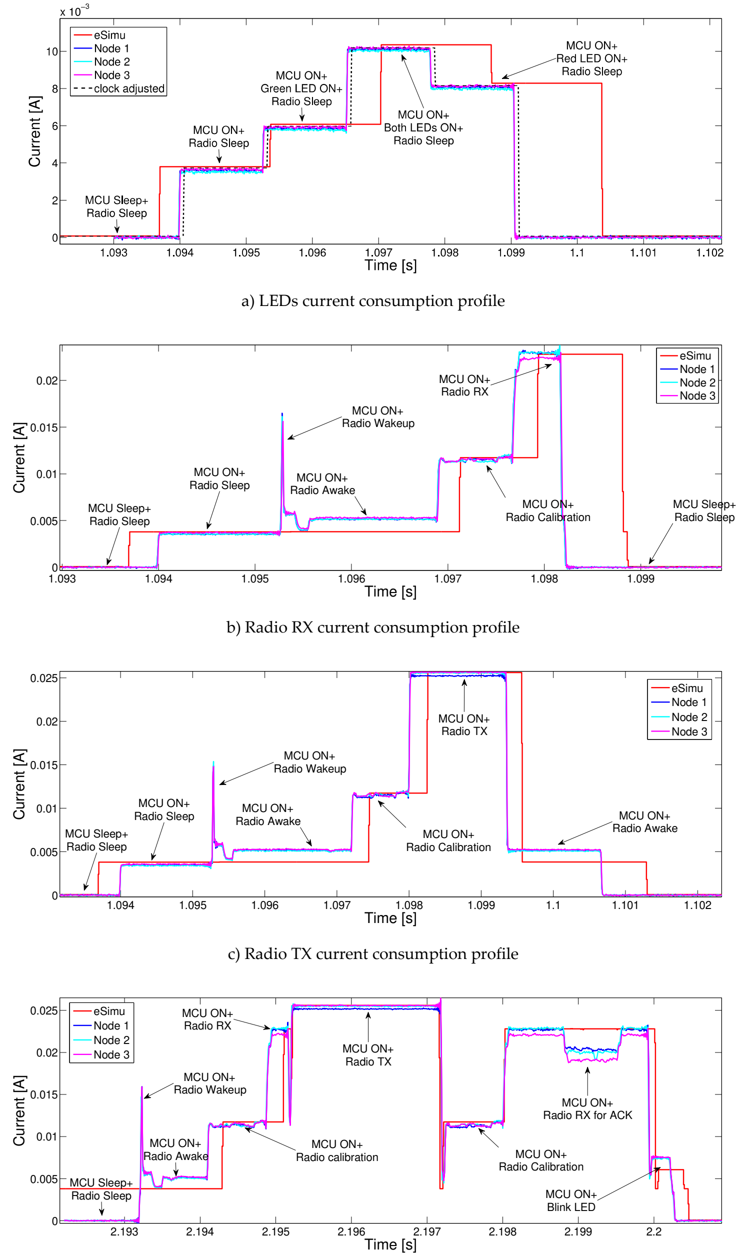
The ez430-RF2500 is a development tool for WSNs produced by Texas Instruments, which includes a MSP430 low-power microprocessor and a CC2500 radio module. External sensors can be connected to the board through a Serial Peripheral Interface (SPI) port, an I2C interface, or a 10 bit Analog to Digital Converter (ADC) with a sampling rate of 200ksps. The standard operational voltage of the platform is 3.6V, by factory the ez430-RF2500 board is powered by an external battery pack (2 AAA batteries). The platform core is an ultra low power micro-controller of the MSP430 family with 32kB of ROM, 1kB of RAM and a clock up to 16MHz [40]. Communication between nodes has been implemented using the CC2500 radio module. The CC2500 radio module is a 2.4 GHz transceiver with a low power hardware wake up function. The transceiver embeds a highly configurable baseband modem, which supports various modulation formats and a configurable data rate up to 500kBd. The Radio Frequency (RF) module does not support Offset Quadrature Phase-Shift Keying (O-QPSK) and for this reason is compliant to the 802.15.4 IEEE standard. Thus it is not possible to implement the ZigBee specification in this specific platform. Texas Instruments developed an alternative network protocol, similar to ZigBee, called SimpliciTI. The SimpliciTI network protocol is a low-power radio-frequency protocol targeting simple, small RF networks whit less than 100 nodes. The SimpliciTI network protocol is designed for easy implementation with minimal micro-controller resource requirements, and supports End Devices in a peer-to-peer network topology, also permitting the usage both of an Access Point to store and forward messages to a LAN, and of Range Extenders, that may extend the range of the network up to four hops [41].

5.2. Current consumption experiment and simulations

In Figure 10 several measurements along with simulation are presented in order to highlight the energy consumption of the examined board in different scenarios. The real current consumption profile has been measured for three different ez430-RF2500 nodes. The measurements have been performed using the shunt resistor technique, as presented in the section 4.1, at V_0=3.3 V using as voltage measurement system a National Instruments USB-4432 Data Acquisition System (DAS). The presented setup guaranties a quantization step of 4.77 μV, given the 24 bits resolution of the DAS and its voltage range (±40 V), resulting in 0.477 μA current resolution with a 10 Ohm resistance. The simulated data has been produced using the Worldsens simulation Framework, presented in section 4.2, applying some modification in order to better model the ez430-RF2500 platform. Such modifications consist in:

- A better model of the platform including as actuator the two LEDs present on the board (each one with different typical current consumption).
- An optimized energy analysis function capable of analyzing trace execution file with high resolution for long execution periods.
- A scale factor to match correctly the execution time; the simulated clock doesn't match the real system clock.
- An energy evaluation of the radio from sleep to wake-up transition.

In Figure 10.a the current consumption of the platform with the LEDs in use is represented. As evident there is a good agreement between the simulated data and the experimental results. In this scenario the platform starts from a condition where booths LEDs are turned on and the Micro-Controller Unit (MCU) in low power mode (LPM3). Then the boards wake up and turn on the LEDs, finally going to sleep again. Notice that the dashed black line represents the simulated current consumption applying the suggested clock adjustment and the solid red curve without the clock adjustment.



a) LEDs current consumption profile

b) Radio RX current consumption profile

c) Radio TX current consumption profile

d) TX with ACK request current consumption profile

Figure 10. Current consumption experimental measurements and simulation results

The current consumption contribution of the RF module is evident in Figure 10.b and 10.c. In those scenarios the platform starts from a LPM3 and after the wake up of the microcontroller it turns on the radio module to perform respectively a reception (RX) and a transmission (TX). It is evident that in such condition the RF module consumes the greatest part of the energy. Even in this case there is a good agreement between experiment and simulation, however there is evident a mismatch between simulations and measurements during the wake up of the radio: The spike measured is due to the current absorption by the crystal oscillator and the WSim software does not take into account such analog phenomena.

The last considered scenario (represented in Figure 10.d) is a bidirectional communication between two nodes using the SimpliciTI network protocol. The node starts in a sleep state, and then it wakes up, turns on the radio, and sends a 40 *char* message to the Access Point (AP) using the SimpliciTI function *SMPL_SendOpt*. The message is sent requesting an acknowledgment in order to confirm a successful transmission. To accomplish the transmission with the acknowledged reception, the RF module switches from the IDLE state to RX state to check if the channel is free, and then starts the transmission. After its completion the node reverts to the RX state, repeating the calibration step and wait for the acknowledgment. Once received the LED blinks to notify the reception and the MCU returns to sleep mode.

6. Conclusions

A review on power consumption measurements in WSN networks has been presented, highlighting the main WSN features, the node architecture, and the network operation. Measurement and simulation techniques adopted to assess the power consumption of a WSN node have been discussed, showing the most significant approaches, the underlying tradeoffs of each methodology, and discussing the achievable accuracy. A case study has been introduced, presenting a characterization procedure and developing improvements for an existing WSN simulator.

Author details

Antonio Moschitta[1] and Igor Neri[2*]

*Address all correspondence to: igor.neri@nipslab.org

1 Department of Electronic and Information Engineering, University of Perugia, Perugia, Italy

2 NiPS Laboratory, Department of Physics, University of Perugia, Perugia, Italy

References

[1] IEEE 802.15TG4 Web site, Available: http://www.ieee802.org/15/pub/TG4.html – Consulted December 2012

[2] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators". IEEE Standards Association. doi:10.1109/IEEESTD.2007.4346346. 2009 – Consulted December 2012

[3] Camilo, Tiago, et al. "An energy-efficient ant-based routing algorithm for wireless sensor networks." Ant Colony Optimization and Swarm Intelligence(2006): 49-59.

[4] Chiti, Francesco, et al. "Energy efficient routing algorithms for application to agro-food wireless sensor networks." Communications, 2005. ICC 2005. 2005 IEEE International Conference on. Vol. 5. IEEE, 2005.

[5] Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." Computer communications 30.14 (2007): 2826-2841.

[6] Tayal, Animesh R., N. V. Choudhary, and Madhuri A. Tayal. "Simulation of sensor nodes for energy conception in wireless sensor networks using finite automata." Proceedings of the International Conference on Advances in Computing, Communication and Control. ACM, 2009.

[7] Barboni, Leonardo, and Maurizio Valle. "Experimental analysis of wireless sensor nodes current consumption." Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on. IEEE, 2008.

[8] Tennina, Stefano, et al. "Locating zigbee® nodes using the ti® s cc2431 location engine: a testbed platform and new solutions for positioning estimation of wsns in dynamic indoor environments." Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, 2008.

[9] Aamodt, K. "CC2431 location engine–Application note AN042." Chipcon products for Texas Instruments (2008): 20.

[10] Orfei, F. et al. "HYBRID AUTONOMOUS TRANSCEIVERS", EDERC2012: 5th European DSP Education and Research Conference, 13/09/2012, Amsterdam, (2012)

[11] Seah, Winston KG, Zhi Ang Eu, and Hwee-Pink Tan. "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges." Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on. IEEE, 2009.

[12] Vullers, Ruud JM, et al. "Energy harvesting for autonomous wireless sensor networks." Solid-State Circuits Magazine, IEEE 2.2 (2010): 29-38.

[13] Potdar, Vidyasagar, Atif Sharif, and Elizabeth Chang. “Wireless sensor networks: A survey.” Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on. IEEE, 2009.

[14] Farooq, Muhammad Omer, and Thomas Kunz. “Operating systems for wireless sensor networks: a survey.” Sensors 11.6 (2011): 5900-5930.

[15] G. Gupta and M. Younis, “Load-balanced clustering of wireless sensor networks”, in Proceedings of 2003 IEEE International Conference on Communications (ICC’03), Anchorage, AK, May 2003, pp. 1848-1852.

[16] R. Rajagopalan and P. Varshney, “Data-aggregation techniques in sensor networks: A survey”, IEEE Communications and Surveys and Tutorials, vol. 8, no. 4, 4th Quarter 2006, pp. 48-63.

[17] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks”, Computer Communications, vol. 30, nos. 14-15, Oct. 2007, pp. 2826-2841.

[18] C. Srisathapornphat and C. C. Shen, “Coordinated power conservation for ad-hoc networks”, in Proceedings of 2002 IEEE International Conference on Communications (ICC’02), vol.5, New York, Apr.-May 2002, pp. 3330-3335.

[19] C. Imrich, H. Kim, and S. Ha, “Dynamic voltage scheduling technique for low-power multimedia applications using buffers”, in Proceedings of the International Symposium on Low Power Electronics and Design, Huntington Beach, CA, June-July 2001, pp. 34-39.

[20] A. Sinha and A. P. Chandrakasan, “Dynamic power management in wireless sensor networks”, IEEE Design and Test of Computers, vol. 18, no. 2, Mar.-Apr. 2001, pp. 62-74.

[21] P. Karn, “MACA - A new channel access method for packet radio”, in Proceedings of the ARRL CRRL Amateur Radio 9th Computer Networking Conference, Redondo Beach, CA, Apr. 1990, pp. 134-140.

[22] V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, “MACAW: A media access protocol for wireless LANs”, in Proceedings of ACM SIGCOMM’94, London, UK, Sept. 1994, pp. 212-225.

[23] J. Monks, V. Bharghavan, and W. M. Hwu, “A power controlled multiple access protocol for wireless packet networks”, in Proceedings of IEEE INFOCOM’01, vol. 1, Anchorage, AK, Apr. 2001, pp. 219-228.

[24] Lattanzi, Emanuele, and Alessandro Bogliolo. “VirtualSense: A Java-Based Open Platform for Ultra-Low-Power Wireless Sensor Nodes.” International Journal of Distributed Sensor Networks 2012 (2012).

[25] A. Bogliolo, E. Lattanzi, and A. Acquaviva. “Energetic sustainability of environmentally powered wireless sensor networks”. In PE-WASUN ‘06: Proceedings of the 3rd

ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks, pages 149– 152, 2006.

[26] E. Lattanzi, E. Regini, A. Acquaviva, and A. Bogliolo, “Energetic Sustainability of Routing Algorithms for Energy-Harvesting Wireless Sensor Networks”, Elsevier Computer Communications, Special Issue on Network Coverage and Routing Schemes for Wireless Sensor Networks, Vol. 30, No. 14-15, pp. 2976-2986, 2007.

[27] Abuarqoub, Abdelrahman, et al. “Simulation Issues in Wireless Sensor Networks: A Survey.” SENSORCOMM 2012, The Sixth International Conference on Sensor Technologies and Applications. 2012.

[28] Hergenröder, Anton, Joachim Wilke, and Detlev Meier. “Distributed energy measurements in WSN Testbeds with a sensor node management device (SNMD).” Architecture of Computing Systems (ARCS), 2010 23rd International Conference on. VDE, 2010.

[29] D. Macii, D. Petri, “An Effective Power Consumption Measurement Procedure for Bluetooth Wireless Modules,” IEEE Transactions on Instrumentation and Measurements, Vol. 56, no. 4, August 2007.

[30] T. Laopoulos, P. Neofotistos, C. A. Kosmatopoulos, and S. Nikolaidis,” Measurement of Current Variations for the Estimation of Software-Related Power Consumption,” IEEE Transactions on Instrumentation and Measurements, Vol. 52, no. 4, August 2003.

[31] Egea-Lopez, E., et al. “Simulation tools for wireless sensor networks.”Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS’05). 2005.

[32] Bich, Walter, Maurice G. Cox, and Peter M. Harris. “Evolution of the'Guide to the Expression of Uncertainty in Measurement'.” Metrologia 43.4 (2006): S161.

[33] Chang, Naehyuck, Kwanho Kim, and Hyung Gyu Lee. “Cycle-accurate energy measurement and characterization with a case study of the ARM7TDMI [microprocessors].” Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 10.2 (2002): 146-154.

[34] Hergenröder, Anton, Jens Horneber, and Joachim Wilke. “SANDbed: A WSAN Testbed for Network Management and Energy Monitoring.” Hamburg, Germany, Aug 8 (2009).

[35] Titzer, Ben L., Daniel K. Lee, and Jens Palsberg. “Avrora: Scalable sensor network simulation with precise timing.” Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on. IEEE, 2005.

[36] Haas, Christian, Joachim Wilke, and Viktor Stöhr. “Realistic simulation of energy consumption in wireless sensor networks.” Wireless Sensor Networks(2012): 82-97.

[37] G. Chelius, A. Fraboulet, E. Fleury, ”Worldsens: development and prototyping tools for application specific wireless sensors networks”, in International Conference on Information Processing in Sensor Networks (IPSN) (Boston, USA), ACM, April 2007.

[38] N. Fournel, A. Fraboulet, and P. Feautrier, “Embedded Software Energy Characterization: using non-intrusive measures to annotate application source code”, Journal of Embedded Computing, Volume 3 (3), IOS Press, July 2009

[39] A. Dwivedi and O. Vyas, “An Exploratory Study of Experimental Tools for Wireless Sensor Networks”, Wireless Sensor Network, Vol. 3 No. 7, 2011, pp. 215-240. doi: 10.4236/wsn.2011.37025.

[40] Texas Instruments, eZ430-RF2500 Development Tool User Guide, Literature Number: SLAU227E September 2007 – Revised April 2009

[41] Texas Instruments, “SimpliciTI Compliant Protocol Stack.” 2009207210. http://www.ti.com/tool/simpliciti – Consulted December 2012.