An advanced study of energy consumption in an IEEE 802.15.4 based network: everything but the truth on 802.15.4 node lifetime

Nicolas Fourty*, Adrien van den Bossche**, Thierry Val**
* LCIS ; Université de Grenoble
51 rue Barthélémy de Laffemas, 26901 Valence Cedex 09
fourty@iut-valence.fr
** CNRS-IRIT-IRT ; Université de Toulouse, UT2
1 place George Brassens, Blagnac 31703 Toulouse, France
{bossche,val}@irit.fr

Corresponding author: Pr. Thierry VAL
email: val@irit.fr
tel: +33(0)562747584 fax: +33(0)562747576

Abstract–*In this paper, the authors present a global study on energy considerations in the context of a wireless network based on the well-known IEEE 802.15.4 technology. Various tests have been realized on a node prototype powered by Lithium batteries in order to determine the lifetime of this network element, in several cases including 802.15.4 coordinator or end device configurations but also non-802.15.4 protocols. The results show that the lifetime is clearly dependent of the node time in the receive state, which has a big impact on not only the network performance like throughput and latency but also the energy budget of the battery. The paper also illustrates the non-linear characteristic of the battery capacity. Finally, it identifies drawbacks which may be avoided in the future design of MAC (Medium Access Control) protocols in order to maximize node lifetime.*

1. Introduction

Nowadays, wireless communications are fully integrated in our daily lives. This is true not only for short range and low throughput communications (remote control, Bluetooth audio streaming and synchronization, ZigBee domestic applications such as lighting) but also for high range and high throughput communications (WiFi domestic networks, WiMAX used for wide range Internet covertures, GSM, GPRS and 3G networks, soon 4G and LTE…). The consumer market is not the only market targeted by the Wireless communication device manufacturers, the industrial area, with much more constraints, are also massively using wireless networks for their field busses replacement or extension. One great advantage of these immaterial links (most often radio communications) is to be completely freed from wires. This advantage comes with their natural drawback which is autonomy and equipment lifetime since they are disconnected from permanent power sources [1] [2] [3]. In order to deal with this point, most of the time we use batteries and accumulators. This is the only choice if you want complete wireless communications such as mobile applications for instance.

In the command/control field for domestic and industrial applications, the ZigBee WPAN [4], based on the IEEE 802.15.4 standard [5] [6], is currently the best suited standard. Indeed, many companies use dedicated wireless equipment, powered from batteries, for low cost data gathering from sensors. These wireless sensor nodes don't have important amount of data to transmit (few kb) and most of the time don't need an important throughput for real time processing. These characteristics are the key of sensor node lifetime optimization. It is essential for the wireless network to keep nodes alive. Considering this, the 802.15.4 standard defines protocols to save energy. However, these protocols are not always implemented due to hardware industrial cost or software limited primitives.

The aim of this work is to accurately study energy characteristics of real components, to identify technological barrier and to propose development advice for future data communication protocols based on the same (802.15.4) physical layer. We will see that some usual choices are clearly not well suited since the radio stage consumes more in the receive mode than in the transmit mode. Announces and data from manufacturers should also be considered with caution since the best performance is often presented.

This paper is divided in four parts: after the introduction, the second Section deals with 802.15.4 principles and main protocols in order to define all the key points and vocabulary. Section 3 presents hardware solutions available on the market and the one we have chosen for our study. The fourth Section focuses on an energy consumption study on a typical lithium (Li-ion) battery. The last Sections are the main contribution and presents metrology experiments with several protocols on the same device dedicated to energy consumption metering: the Section 5 presents some very simple transmission protocols experiments while 802.15.4 MAC protocol energy budget is studied in Section 6. At last, conclusion and perspective are discussed.

2. IEEE 802.15.4 standard overview

This Section proposes an overview of the IEEE 802.15.4 standard Medium Access Control (MAC) Layer. The goal of this Section is to remind the basics of the technology in order to understand the next part of the article.

2.1. Generalities

IEEE 802.15.4 [5] [6] [7] [8] is a standard which proposes an original two-layer protocol stack with PHYsical layer (PHY) and a Medium Access Control layer (MAC) for Low-Rate Wireless Personal Area Networks (LR-WPAN). Typical targets are embedded communication systems with very low power consumption [9]. It proposes a lightweight protocol stack for applications which require low data rates (up to 250kbits/s in the original version of the standard) and low latency. Typical applications of this technology are Wireless Sensor Networks (WSN), which can be used either in industrial, domestic or medical contexts. Moreover, the IEEE 802.15.4 standard is promoted by the ZigBee Alliance [10] as the physical layer and data-link layer of ZigBee Network specifications.

An IEEE 802.15.4 network is built using several types of devices, including end devices, coordinators and PAN coordinator. The standard describes two protocol stack versions: first, there is a Full Function Device (FFD) stack, including all network standard functionalities, such as sending/receiving data, routing and coordinating. Then, there is a lighter version or Reduced Function Device (RFD) stack. This stack is used by end devices. With this stack, several functionalities can be removed, like routing functions, data encryption, etc. An RFD stack requires little memory and CPU.

One of the particularities of most IEEE 802.15.4 hardware is the possibility of total reprogrammation of the modules, including the MAC layer, which is not often possible for example, in the 802.11 context. This possibility is particularly interesting for developing and evaluating the performance of original MAC layer protocols [11] [12] [13].

2.2. IEEE 802.15.4 PHY Layer

The IEEE 802.15.4-2003 [5] and 802.15.4-2006 [6] standards propose several physical layers (868MHz, 915MHz or 2.4GHz). The most commonly used is the PHY2450, which operates on the 2.4GHz radio frequency band. Many new PHY layers based on CSS (Chirp Spread Spectrum) and UWB (Ultra Wide Band) are proposed in the last standard amendment [7] but UWB or CSS transceivers are not yet fully stabilized and not available at this time at a reasonable price. The next Sections of the paper are based on the 802.15.4-2006 PHY.

All nodes from an IEEE 802.15.4 network use the same radio channel. Protocols are optimized for short and periodical data transfers; nodes usually stay in a sleeping mode, called the *doze mode*. In this mode, all radio functionalities are switched off, removing the ability to receive messages. The waking time has to be set before going into the doze mode (synchronous wake-up), but sleeping devices may also wake up if a local event occurs (asynchronous wake-up), for example, sensor detection. On the other hand, network coordinators that centrally manage communications always remain awake to buffer messages for sleeping terminals.

2.3. IEEE 802.15.4 MAC Layer

The IEEE 802.15.4 MAC protocol has been optimized for Low Power consumption. It features two operating modes: a non beacon-enabled mode and a beacon-enabled mode.

In the non beacon-enabled mode, nodes access the channel using a non-slotted CSMA/CA mechanism. In beacon-enabled mode, time is subdivided in superframes which always begin with a beacon frame sent by the coordinator. The superframe is defined by the Beacon Interval (BI) and the superframe duration (SD). The BI defines the time between two consecutive beacon frames. The SD defines the active period in the BI. Optionally, an inactive period is defined if BI>SD and the nodes may enter sleep mode to save energy during the inactive period. The duration of BI and SD depends on a constant *aBaseSuperframeDuration* defined in the standard and two parameters, the Beacon Order (BO) and Superframe Order (SO), according to the relations (1) and (2) where 0≤SO≤BO≤14.

BI = *aBaseSuperframeDuration**2BO (1)

SD = *aBaseSuperframeDuration**2SO (2)

During each superframe, the active period is equally divided into 16 slots that form two different periods with two different medium access mechanisms: Contention Access Period (CAP), where nodes access the medium using slotted CSMA/CA, and Contention-Free Period (CFP), where the medium access is regulated by the Guaranteed Time Slots (GTS) mechanism. The latter mechanism is the more suitable for real-time traffic since timeslot allocation suppresses frame collisions.

In beacon-enabled mode, medium access is done with the slotted CSMA/CA mechanism, the slot boundary is determined via the beacon synchronization. This mode enables deep energy saving without loss of connectivity thanks to periodical wake up. In this case, end devices wake up before the beacon, get data pending and go back in sleep mode, minimizing activity.

Optional GTS mechanism is available to support time-constrained traffic. GTSs enable end-devices to get a timeslot where frames are transmitted without collision. End-devices request to their coordinator for a GTS which grant is periodically announced in the coordinator's beacon.

3. IEEE 802.15.4 hardware solutions

While we were investigating on 802.15.4 hardware solutions, we were confronted to the very small number of transceiver hardware available. Moreover most solutions used the same hardware architecture with the processor (MCU) communicating with the transceiver through an SPI communication port. The transceivers from Freescale (1321x, 1322x, 1323x), Texas (CC2420, CC2520, CC2530…) or Ember/ST (Maxstream XBee version 2) are the most implemented on hardware solutions. They are 802.15.4 compliant which enable manufacturer to design their own solution by combining a processor and a transceiver according to the computational power needs of the application. Some UWB/CSS 802.15.4a PHY compliant transceivers will be available in a near future.

3.1. Different power conservation modes for different applications

Some applications are focusing on the computation power such as multimedia sensor networks [14]. These solutions would prefer a 32-bit processor such as ARM7 on Freescale 1322x, or Intel PXA271 Xscale on Crossbow Imote2. These hardware solutions often use a proprietary software stack based on the OS designed to fit the processor. However recent advances in this field have raised the interest of open operating system such as the TinyOS, the RTOS or ContikiOS. Except for the Freescale 1322x, which has dedicated hardware for dumping RAM in the transceiver, the weak point of these solutions is the data transfer from the processor to the transceiver. Indeed, the SPI communication port limited to 500kbit/s is often the bottleneck of such systems. Manufacturers are trying to reduce the latency of the data transfer by using dedicated hardware which is really interesting for delay sensitive applications and WMSN. The counterpart of this additional hardware is the increase in the energy cost of the node.

On the other hand some applications are focusing on node lifetime and are less interested in computational power such as industrial sensor network. These solutions would prefer an 8-bit processor such as 1321x from

Freescale which uses MC9S08 Freescale processor or a 16-bit processor such as Telosb from Crossbow which uses MSP430 Texas processor. These motes don't need a great amount of computational power and are already implemented in real industrial applications [15]. Moreover these systems don't need a full operating system and the software is often simple code written in C language. Compared to operating systems using a task scheduler, task management on these systems is more complex due to limited internal components (RAM, Flash…). However the gain in power dissipated between an 8-bit processor module and a 32-bit processor module can be more than a factor 10! While the idle power of the Freescale 1321x is only 3mA, the idle power consumption of the Crossbow Imote2 module is 35mA. The remaining of the paper will be focused on energetic concerns and node lifetime maximization.

The three operation modes *Doze*, *Idle* and *Hibernate* are implemented in all 802.15.4 compliant current platforms. The switching between these modes enables the platform to save energy by powering on and off electronic components on the platform. While on the 32-bit processor Imote2, a specific component is in charge of energy management, on the 8-bit processor modules the switching between modes is done by powering on and off the MCU and the transceiver. As a consequence several power levels are possible just for Idle and Hibernate modes: MCU ON / Radio ON (Idle mode), MCU ON / Radio OFF (Hibernate mode with MCU), MCU OFF/ Radio OFF (Hibernate mode) these states are macroscopic views because MCU has often several level low power modes.

For instance, the Doze mode of our prototype node presented in Section 3 is a particular case of the combination presented above: the node going to Doze mode puts the Radio in a low power mode for a fixed amount of time. As consequences, the Radio stage giving the main clock to the MCU, it is stopped but powered, enabling a quick restart as soon as the transceiver wakes up and supplies the clock signal again.

The energy profile switching is the key parameter of 802.15.4 based networks. In order to control sharply this profile switching, Freescale gives for their platforms a set of commands enabling software designer to handle these profiles. This set of commands called SMAC (*Simple Media Access Controller*), which is open source, will be used by the software on our prototyping node described in the next Section.

3.2. Description of our prototype node

3.2.1. Objectives

Although several out of the box solutions are available from Crossbow Imote2 or Freescale 1321-SRB, these boards often implement other functionalities such as a USB driver, a UART level converter or sensors which can drain energy. That's why we designed our own prototype using hardware solutions previously described with the following constraints:

- We wanted a minimal board in terms of components (no UART, no USB, only one sensor powered from the MCU), in order to maximize energy savings.
- We didn't want any energy converter (no LDO, no DC/DC converter), in order to prevent conversion loss. We also wanted to use consumer products like AA standard batteries.
- We wanted to spare components powering, in order to monitor energy consumption of each electronic block (MCU, Transceiver, Sensor, RF switch, LED)
- We wanted an integrated battery voltage measurement in order to monitor the battery voltage evolution and the impact of radio pulses on batteries.
- We wanted a completely open software, giving full access to all the energy saving profile combinations and giving us the possibility to build our own MAC protocols
- And at last we wanted a small size prototype with a lot of programmable input/output available for future applications.

All these constraints explain our choice of the Freescale ZRD01 [16]. This platform is composed of a MC9S08GB60 associated to the 13192 transceiver. For the next generations 1320x and 1321x Freescale have embedded the platform in a System in Package which makes it more difficult to gather some measure on power consumption of each electronic block. For more flexibility the ZRD01 has been soldered to a daughter board. This daughter board contains the RF antenna and route some pins of the ZRD01 to the motherboard. The motherboard contains: the BDM programming port, the integrated battery voltage measurement, a temperature sensor (LM61BIZ from National Semiconductors), 2 push buttons for reset and debug, 8 PIO, 1 LED and test points for measuring current consumption. This hardware associated to the SMAC open stack satisfies all of our constraints. Figure 1 presents our prototype node.

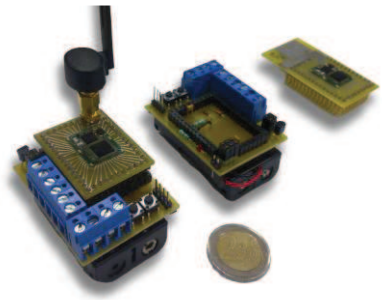

Figure 1. Our prototyping node

3.2.2. Our node characteristics

The aim of this measurement campaign was to check the commercial datasheet of the module provided by the manufacturer [16]. Moreover for a deep energetic consumption investigation we have split the work in two parts. The first part of the campaign was focused on current consumption for each electronic block. We have listed 5 main blocks: Processor, Transceiver, Integrated Voltage Measurement, Sensor and LED (no PIO used). For each of these block we measured current consumption for each protocol state defined.

The second part of the study was focused on the duration of each protocol state. For this part we measured time constraints for each step of the protocol: *Transmission* (TX), *Reception* (RX), *Sensor capture*, *Data processing* and *Idle*. For the rest of the Section we will consider the Lithium Batteries having a fixed Voltage over the duration of measurements. This assumption, justified by the next Section, will enable us to simplify our power consumption measures. Moreover we will consider the current consumption is constant over a protocol state. This assumption fits our measurements as in Figure 2.

3.2.3. Prototype energetic budget

The measures presented in this Section have been taken directly on the prototype and obtained from: µA-meter, digital oscilloscope and ANT16 logic analyzer. Results are presented in Table I and Table II.

On Figure 2, the first channel, above, represents the MCU current consumption while the second channel, under, represents the transceiver current consumption. The current is obtained by measuring the voltage across a 10Ω resistor. For this example, we have chosen a typical wireless sensor network application cycle: the mote awakes from the macroscopic Doze mode, then goes to Rx Mode waiting for a beacon, then returns to idle mode to collect data from the sensor and then sends the data (19 bytes) to the coordinator. On the figure the colors represent the macroscopic modes defined by the 802.15.4 standard (Idle, Rx, Tx, Doze).

We check on this figure that our assumption that currents over the macroscopic states are constant is verified. There is three noticeable events noted A, B and C on the figure.

The A event is due to the main clock switching. While in the Doze mode, the node uses the internal oscillator for lower power consumption (Real Time Clock), when the node awakes the main clock is switched to a high frequency oscillator for faster operations but also a bigger consumption.

The two B events represent Rx transient states. Two cases are presented a startup time for the radio and an overlapping period where MCU is activated and the Radio stage is not idle yet. These states are mentioned in Table III as “other states” (A and C events are measured in the same way).

The C event represents a Tx transient state due to transmission level startup (transmission power is 3.6dBm on the figure).

These three noticeable events have a negligible impact on the current consumption of the device (cf Figure 6).

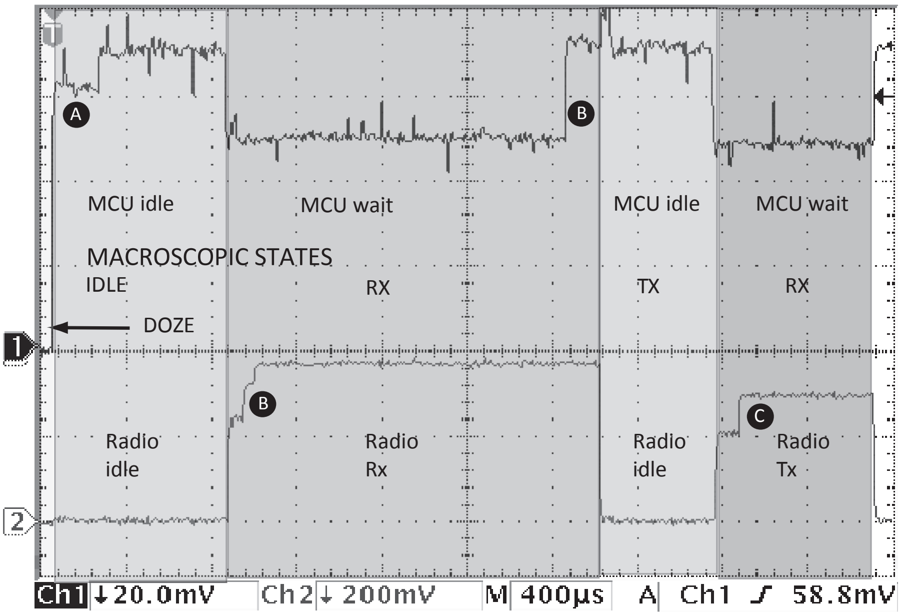

Figure 2. Current consumption for MCU and transceiver

TABLE I

CURRENT CONSUMPTION IN EACH STATES FOR EACH ELECTRONIC BLOCK

States	MCU	Tran.	T° sensor	Batt. sensor	LED
Tx (0 dBm)	3.5 mA	30 mA			
Tx (3.6 dBm)	3.5 mA	37 mA			
Rx	3.5 mA	38 mA			
Capture T°	5 mA	1.3 mA	120 µA		
Battery monitor	5 mA	1.3 mA		1 mA	
LED_ON	5 mA	1.3 mA			700 µA
Sleep	140 µA				

It is important to notice that on our prototype, the time required to send or to receive one byte of data is the same and is equal to 48.36µs. Indeed data is first transferred to the transceiver by the SPI (16µs/Byte) and then transmitted on the medium at the rate specified by the 802.15.4 standard (32µs/Byte). These first measures enable us to associate an energy cost to each action of the prototype node.

TABLE II

REQUIRED TIME FOR EACH STEP (INCLUDING MCU PROCESSING TIME)

Action	Req. Time
Sense T°	20 µs
Sense battery	20 µs
Transceiver wake-up	310 µs
MCU wake-up	235 µs
Idle to sleep mode for MCU and transceiver	570 µs
Transmission of *n* bytes	(423+48.36**n*) µs
Reception of *n* bytes	(289+48.36**n*) µs

As we can see on Tables I and II, current consumption and timings of the node prototype have been identified. Theses values can be used as parameters for an energy model inside a network simulation tool. This work has been done in our team with the OPNET simulator in a Wireless Sensor Network simulation based on the [17] model. The results are interesting since simulation converge with prototype.

4. A typical battery study

Nearly all 802.15.4 compliant hardware solutions are equipped with low voltage detector, enabling the system to send an alarm when the battery voltage reaches a threshold. Some systems even go further, including a battery monitoring system. However these systems often use models dedicated to alkaline batteries when, in a WSN context where the node lifetime is critical, we would like to have the largest battery capacities.

The best solution for powering a WSN node is lithium batteries. Indeed, the large capacity of lithium batteries enables us to neglect the rate capacity effect due to high current needed by the system. Other advantages of lithium batteries compared to alkaline batteries are their very low self-discharge rate, and their very stable voltage. That’s why we have selected a Lithium AA type battery [18].

As described above, most of discharge models commonly used are not suited for this kind of batteries and although several industrial discharge profiles are given on the datasheets we have performed several tests in order to verify the battery capacity in a WSN application context [19] [20].

4.1. Battery characterization platform

The battery test and calibration hardware platform is based on the Freescale 13213 SRB evaluation board. This board uses the MC9S08 Freescale microcontroller and implements a temperature sensor and a 10 bit Analog to Digital Converter (ADC) which has been used to monitor the battery voltage during the experiment. The Figure 3 shows the electrical schematics.

On the Figure 3 and Figure 4 the resistance R enables us to fix the discharge current of the battery. The 13213 module, powered from the PC USB Port, has two functions. The first one is to monitor the battery discharge using one of its ADC channel and one of its Programmable Input/Output (PIO) commanding the discharge through an NMOS transistor. The second function of the module is to transmit data acquired using a virtual UART on the USB link. Once the data is acquired by the PC, data is processed and stored in a file until the battery fails.

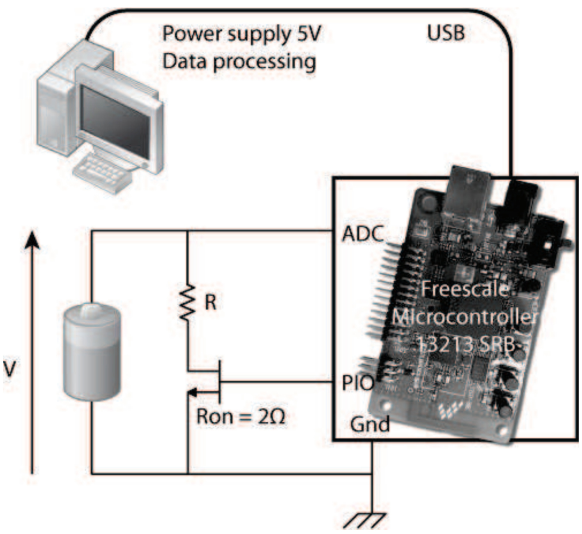

Figure 3. Electrical schematics of the battery characterization platform

The ADC is configured with a fixed 3.3V reference voltage created by the linear regulator integrated on the Freescale module. The 10 bit ADC enables a resolution of 3mV. During all the experiment, the temperature is monitored and kept constant (25°); thus, temperature variation effects have not been studied.

4.2. Capacity computation

The lifetime depends on several parameters such as battery chemistry, discharge current, temperature, cut off voltage, etc. In this Section, we will vary the discharge current profile. The datasheet specify a battery capacity around 3A.h (considering a cut off voltage of 0.9V).

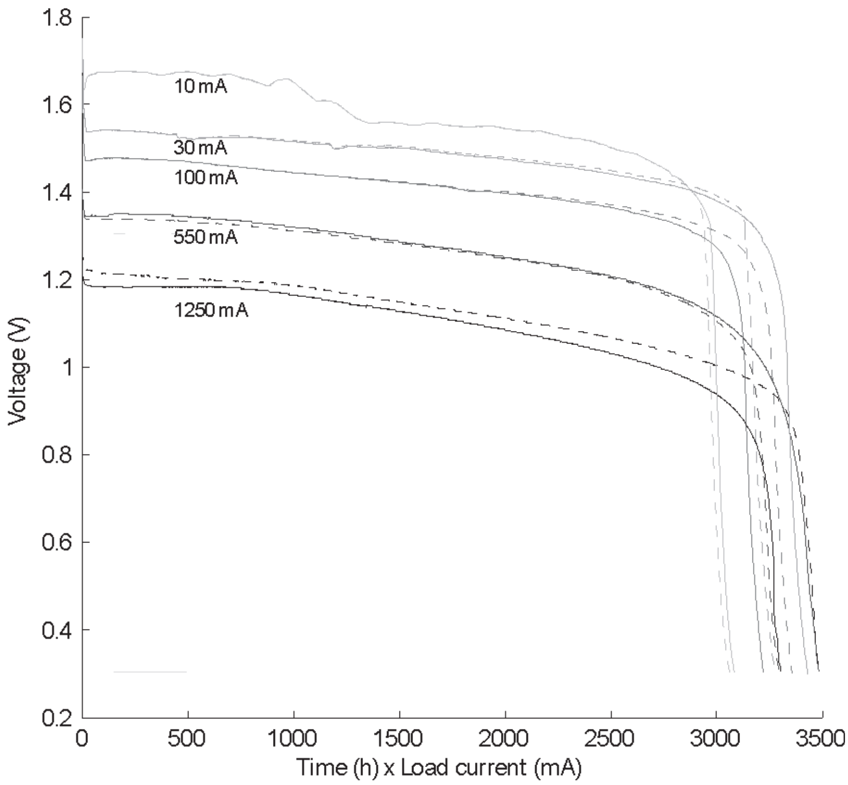

Figure 4. Battery voltage evolution (fixed current)

Real test discharges have been done on the validation prototype previously described and have been configured first to make continuous discharge by fixing the value of the resistor R. Figure 4 represents the battery voltage evolution, when the battery is loaded with a fixed resistor. For better comprehension we have normalized each discharge by its discharge current. Each discharge has been done at least twice. Figure 4 confirms the datasheet information with a total capacity measured a bit better than the capacity announced between 3 and 3.5A.h

In order to go deeper in our investigations, we have made more realistic discharge current for WSN environments. We have created two profiles:

- Profile 1: this profile has been thought to represent the energy consumption of a router node. This profile has a cycle duration of 2s, a duty cycle of 50%: activity 100mA current, then inactivity MOS transistor leakage current (neglected).
- Profile 2: this profile has been thought to represent the energy consumption of a sensor node. This profile has a cycle duration of 2s, a duty cycle of 10%: activity 100mA current, then inactivity leakage current of the MOS transistor (neglected).

Figure 5 presents the comparison of fixed current and WSN representative profiles. Each discharge has been done twice (solid and dotted lines on Figure 5). Even if the 100mA current chosen is not perfectly suited for WSN applications, a 10mA current (more relevant) would have required more than 4 months for each measure.

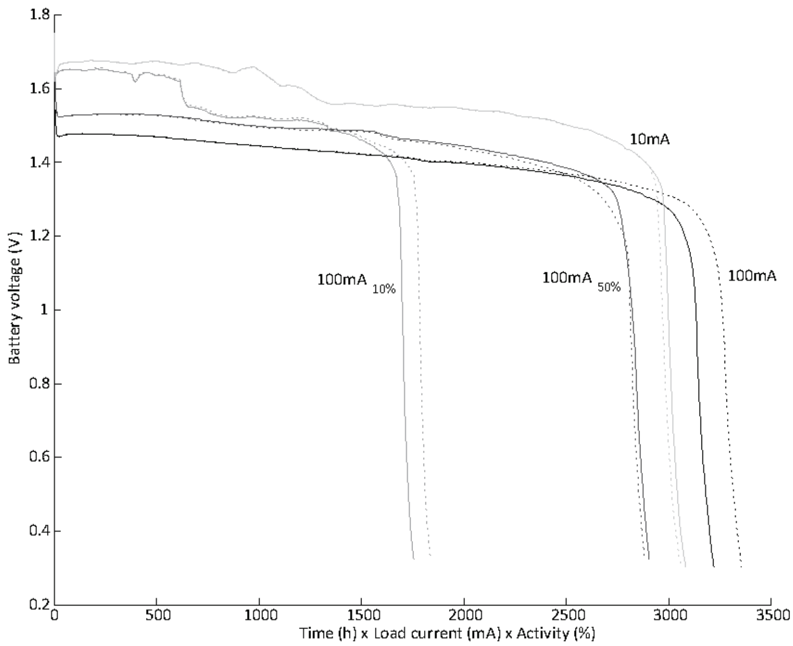

Figure 5. Battery voltage evolution (fixed current and WSN profiles)

Three important points can be noticed on this figure. The first point is that if we compare profile 1 (50% activity) with the fix 100mA load, we can notice a decrease in battery capacity (below 3A.h). The average loss is around 13.3% of the battery capacity. The second point is that this phenomenon also appears with profile 2 (10% activity) and is much more important with a measured capacity of 1.8A.h and a 40% capacity loss. The last point is that if we compare profile 2 with the fix 10mA load, which has the average current as the constant 10mA discharge, the phenomenon is still present with approximately the same loss levels.

These first investigations have led us to pay attention to the load variation, since they may have important negative effects on the battery capacity and in a WSN context on the node lifetime.

In order to measure the impact of the variable loads on real hardware in a WSN context we have preferred not to investigate much more on the duty cycle and the cycle duration. Instead we have focused our efforts on the energy consequences of the MAC mechanisms on our prototype.

5. Some very simple transmission protocols experiments

5.1. Without acknowledgement nor synchronization (maximum lifetime)

Most 802.15.4 transceiver manufacturers announce lifetimes greater than a few years. Our first experiment is to verify the feasibility of such an important lifetime on a real node.

This first experimental protocol (scenario 0) is very simple and detailed hereafter:

- Module wakes-up,
- Sensor acquisition : Sensing temperature then sensing battery voltage,
- Frame set-up: the frame has a fixed size of 19 bytes: 802.15.4 preamble, frame length field, addresses, sensor data formatting, sequence number and a Frame Check Sequence (FCS),
- Frame transmission: done without any channel listening before the transmission (collision possible),
- Return to doze mode for 2.5 seconds.

When the node is not transmitting data, it returns immediately to the doze mode described in Section 3, without waiting for an acknowledgement. The node is powered from 2 AA Lithium batteries presented in the last Section. The sink is a Freescale 13213-NCB reference platform connected to a PC through a serial link for data monitoring and storage, and is always in RX mode. Although this protocol is not representative of a WSN context, this experiment will determine the maximal lifetime limit for a fixed activity rate with this hardware.

With the results presented in Table I and Table II, we are able to compute the energy consumption for each stage of this experimental protocol. Table III presents each stage and the total current consumption for the cycle. The energy is computed assuming that the battery voltage is constant and equal to 3V. This assumption is dependent on the Lithium battery choice presented in Section 4. The energy budget of this first study is presented in Table III, scenario 0.

For an activity rate of 2ms on a cycle of 2.5s (0.08%), the obtained lifetime is 484 days (1 year and 2 months). This first experiment led on the long term has demonstrated the capacity of 802.15.4 transceivers to send data for years with a precise wake-up (*doze* mode). Nevertheless, this protocol is not usable in a real context since the MAC protocol must be taken into account. The next experiments presented in the paper will try to get closer to a real communication protocol.

5.2. With acknowledgement and no synchronization

For this new experiment, the previous protocol is taken and an acknowledgment is added between the frame transmission and return to the doze mode: the node waits for the acknowledgement from the sink (timeout 2ms, 2 retransmissions max). The choice of the 2ms watchdog for the acknowledgment has been fixed by the hardware solution. The SPI bus between the MCU and the transceiver is the bottleneck in terms of delays, however this delay is minimal since the 802.15.4 standard acknowledge is composed of only 3 bytes at the MAC level (and 11 on the medium).

The two extreme cases where studied: without retransmission (scenario 1) and with 2 retransmissions (scenario 2). The energy budgets are presented in Table IV and illustrated with pie charts in Figure 6.

TABLE III
ENERGY CONSUMPTION FOR EACH STATE OF THE EXPERIMENTAL PROTOCOL

States	Time (ms)	Current (mA)	Energy sc.0 (µJ)	%	Energy sc.1 (µJ)	%	Energy sc.2 (µJ)	%
Data Tx	1,341	33,5	134,85	**11.19**	134,85	**9,28**	404.55	**18,21**
ACK	2	41,5	-	-	249	**17,13**	747	**33,63**
Sleep	2500	0,14	1050	**87.18**	1050	**72,25**	1050	**47,27**
Idle to Sleep			15,22	1,26	15,22	1,05	15.22	0,69
Other states			4,27	0,35	4,27	0,29	4.27	0,19
Total			**1204,3**	100	**1453,3**	100	**2221.04**	100

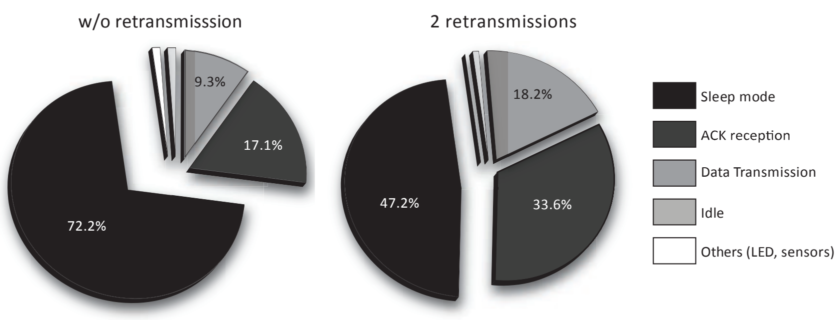

Figure 6. Energy consumption repartition for different scenarii

On Figure 6, two elements are noticeable:

- For a cycle without retransmission (optimistic scenario), the total energy required for one cycle is 1,45mJ and 1,05mJ are required to keep the prototype in Sleep Mode. Indeed 72.2% of the total battery capacity is consumed to do nothing but keeping the node alive in doze mode!

- For a cycle with two retransmissions (pessimistic scenario), the total energy required for one cycle is 2,22mJ. The energy to keep the node alive in doze mode is still considerable with 47.2% of the total battery capacity consumed.

Even if the pessimistic scenario drastically increases the energy budget and decreases the part of the doze mode energy consumption, the doze mode current is a key parameter when designing a WSN focusing on the node lifetime. Although you can save energy using a low throughput 802.15.4 compliant solution and optimizing the protocols, the hardware solution must privilege the ultra low current in doze mode parameter.

At this point, we have investigated only some very simple protocols to send data. We are now going deeper in the networking with a real MAC algorithm – the one proposed in the IEEE 802.15.4 standard – in order to see the node lifetime in a real WSN context.

6. IEEE 802.15.4 MAC protocol energy study

In our last experiment we have seen the impact of retransmissions on the energetic budget, and specifically the impact of the reception window to receive the acknowledgment. This impact increases with the fact that the reception window must be oversized to compensate the non-deterministic behavior of the communication.

In order to study more realistic protocols, we have made several energy studies on the same hardware presented above but this time using WSN regular medium access control algorithms. The aim of this study is to evaluate the node lifetime, the node being powered by the batteries presented in Section 4.

6.1. Study cases

All of our study cases use the IEEE 802.15.4 Beacon-enabled mode presented in Section 2. In this mode the MAC layer uses a superframe structure and alternate activity cycles (computing, transmit, receive) and inactivity periods (stand by). Due to the beacon-enabled mode, the node has to wake up periodically to listen to the beacon from the coordinator which has a significant cost on the node lifetime.

Several experiments have been done using two IEEE 8021.15.4 compliant stacks on the same hardware:

- The first experiment uses the out of the box Freescale IEEE 802.15.4 stack. It is free but not open source and is completely compliant with the hardware. However this stack does not implement the *doze mode*.
- The second experiment uses LiMAC stack (Light MAC), a minimalist and open source stack developed at the IRIT laboratory which fits our hardware; it handles both Hibernate and Doze power saving mode.

Both software solutions have been configured to use the same hardware in order to highlight the energetic impact of power savings modes in a practical experiment.

In order to eliminate random effects (collisions, interferences, etc.) during experiments, we have forced the hardware in the RX state since it is the most energy consuming state. We can consider this situation as a worst case for the energetic study. All the nodes have been powered by two of the selected AA Lithium batteries and their voltage level have been monitored through the battery sensor presented. Then, we have determined the node lifetime considering a cut-off voltage of 1.9V. Under this threshold, the sensor is still alive but has an erratic behavior sending corrupted data or not being able to activate radio stages of the component. Table IV summarizes our profile settings.

TABLE IV
TEST PROFILE SETTINGS

#	Stack	BO	SO	ecoMode	Period (ms)
0	Freescale	7	4	Idle	1966.08
1		4	1	Idle	245.76
2	LiMAC	7	4	Idle	1966.08
3		7	4	Doze	1966.08
4		4	1	Idle	245.76
5		4	1	Doze	245.76

The BO and SO parameters of the 802.15.4 superframe have been chosen in order to have a 8% duty cycle which is a good compromise between the data load and inactivity periods. This duty cycle is relevant for most typical wireless sensor networks and is often used by commercial ZigBee compliant device implementations. :

- The pair BO=7/SO=4 defines a cycle period of 1966.08ms for an active period of 245.76ms (#0, 1, 2 and 3),
- The pair BO=4/SO=1 increases the reactivity of the system and defines a cycle period of 245.76ms for an active period of 30.72ms (#4 and 5).

Moreover these settings enable comparison with the simulated discharges of 10mA (continuous) and 100mA (duty cycle 10%) presented in Figure 5. It is important to note that the Freescale stack does not implement the doze mode and, thus consumes more energy, however the processor is powered and is able to process or to harvest data from sensors. In the doze mode, used to save energy, the MCU is stopped and cannot process any data.

6.2. Experimental results

Table V summarizes our results according to the different profiles. In this Table profiles #1 and 2 are Freescale reference stack configured with the two couples of BO/SO selected above. Since the Freescale stack is given as a binary file, we haven't been able to take advantage of the battery monitoring sensor on the prototype. As a consequence, the average current and the equivalent battery capacity are not available with this stack. The other profiles (#2-5) use LiMAC stack with the BO/SO selected above and using the possibility to put the system in Doze mode in the inactive period.

For each profile we have summarized: the measured Node lifetime in h, the measured Average current in mA, the computed Equivalent battery capacity computed as in Section 4. It is important to note that both the Freescale stack and LiMAC stack configured to have inactivity periods in the idle mode have quite the same lifetime. We assume that the difference is explained by the proprietary implementation of the stack.

TABLE V
NODE LIFETIMES AND EQUIVALENT BATTERY CAPACITIES

#	Stack	BO	SO	ecoMode	Lifetime (h)	Average Current (mA)	Battery Capacity (mA.h)
0	Freescale	7	4	Idle	188	*NA*	*NA*
1		4	1	Idle	188	*NA*	*NA*
2	LiMAC	7	4	Idle	158	9.18	1450
3		7	4	Doze	455	3.292	1498
4		4	1	Idle	166	9.18	1524
5		4	1	Doze	428	3.292	1409

6.3. Discussion

This advanced study on 802.15.4 compliant devices used in a WSN context enables us to discuss real node lifetime.

6.3.1. The truth on 802.15.4 node lifetime

It is important to note that in the beacon-enabled mode, the node has to wake up periodically. This constraint has a tremendous impact on node lifetime: in this mode, the Freescale stack can work for only 8 days, while LiMAC, with doze mode, increases the lifetime up to 19 days.

This result has to be considered as the worst case since for data transmission emulation we only force the RX state on the transceiver which maximizes both TX and idle state. The ecoMode impact can be verified for the two pairs of BO/SO investigated (profiles #0, 1, 2, 4) represented on Figure 7.

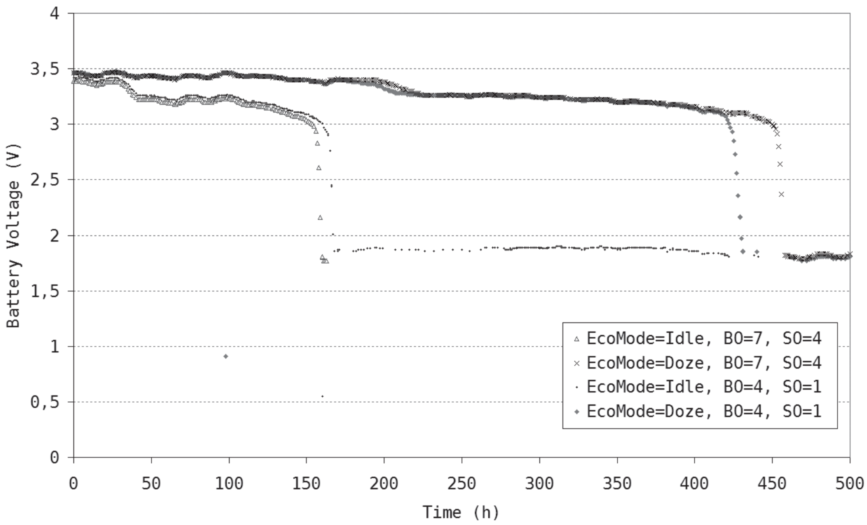

Figure 7. EcoMode impact on node lifetime with LiMAC stack

6.3.2. To doze or not to doze?

If we compare profiles #(2, 4) with #(3, 5) we can notice that the doze mode enables the node lifetime to be multiplied respectively by a factor of 2.88 and 2.58. This lifetime increases at the cost of the processor availability for computation. Since this is a critical point for intelligent systems needing processing capabilities permanently, this energy saving mode will be used or not according to the application.

Another important factor is the current consumption profile switching. While using the Idle Mode as the ecoMode (#2, 4) the duration of active period seems to increase lifetime when the active period decrease. However when the current profiles are really different (Doze used in ecoMode, #3, 5) the lifetime decrease when the active period decrease. This phenomenon is coherent with our assumption of capacitive effects on the battery capacity computation in Section 4.

6.3.3. Duty cycle impact on battery capacity

We have seen in Section 4 that the equivalent battery capacity was reduced as the activity duty cycle was going down. Our profiles implementing an 8% duty cycle, we can compare the equivalent battery capacity computed with the results presented in Section 4, particularly the 10% activity profile. Indeed, the battery capacity reduction is also noticeable on our study cases. Since our study cases have an activity cycle inferior to the 10% activity emulation profile we expected a smaller battery capacity. The results of our experiments confirmed our expectations since we computed an equivalent battery capacity of 1500mA.h which is significantly less than the 1750mA.h of the 10% activity profile results from emulation.

7. Conclusion and future works

In conclusion of this node energy metrology, we have demonstrated that the lifetime announced by hardware manufacturer was clearly not suited when we use devices for networking. Indeed we have first showed the impact of variable loads on battery chemistry reducing the node lifetime. These effects are non linear and are hardly predictable. Even at low duty cycle, the high currents needed for transmit/receive operations reduce drastically the node lifetime. Moreover, the second point discussed was the power consumption during the receive mode. This mode represents the worst case in terms of power consumption. The study cases presented have shown that this mode should be limited for energy purposes and raised an interrogation on the use of CSMA/CA for energy saving. In a third stage we have demonstrated the effectiveness of low power modes for increasing node lifetimes. This effectiveness comes with the sacrifice of processing power for application. Moreover these modes are highly hardware dependant. However we have also demonstrated that long lifetime was possible at the cost of protocol complexity (MAC protocols).

The results presented in this paper enable us to evaluate energetic characteristics of an original WSN MAC developed in the team: values have been introduced as parameters in the WSN node energetic model of IEEE 802.15.4 [17] in the OPNET simulator. A comparison with a real prototype implementation is in progress.

In future works, we plan to continue the simulation works on OPNET and WSIM (specific microcontroller platform simulator) [21] and extend our metrology using both hardware prototype and simulated nodes. This could lead us to implement and demonstrate a consumption model for this node including lithium batteries discharges. We also plan to use and design specific energy efficient MAC algorithm such as S-MAC in order to study interactions between MAC and Network (routing algorithms) protocols [21]. At last we plan to do the same kind energy metrology on next generation 802.15.4 hardware using chirp modulation and localization algorithms.

References

[1] Andrew Y. Wang et al., *Energy efficient modulation and MAC for asymmetric RF microsensorsystems*, IEEE Intl. Symp. Low Power Electronics and Design, 2001, pp. 106–111.

[2] QicaiShi and Edgar H. Callaway, *An ultra-low power protocol for wireless networks*, Proc. 5th World Multi-Conference on Systemics, Cybernetics and Informatics, 2001, v. IV, pp. 321–325.

[3] V. Jelicic, V. Bilas, *Reducing power consumption of image transmission over IEEE 802.15.4/ZigBee sensor network,* Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE pp 1211 - 1215, Austin, TX ISBN: 978-1-4244-2832-8

[4] Andrew Wheeler, *Commercial Applications of Wireless Sensor Networks Using ZigBee*, IEEE Communications Magazine, April 2007

[5] IEEE, IEEE 802.15.4-2003, *Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate WPANs*, 2003

[6] IEEE, IEEE 802.15.4-2006, *Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate WPANs*, 2006

[7] IEEE, IEEE 802.15.4a-2007, *Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate WPANs; Amendment 1: Add Alternate PHYs*, 2007

[8] Jose A. Gutierrez et al. IEEE 802.15.4: *A developing standard for low-power, low-cost wireless personal area networks*, IEEE Network, v. 15, n. 5, September/October 2001, pp. 12–19.

[9] Casilari, E.; Cano-García, J.M.; Campos-Garrido, G. *Modeling of Current Consumption in 802.15.4/ZigBee Sensor Motes*. Sensors 2010, 10, 5443-5468.

[10] ZigBee Specification-Document 053474r17, www.zigbee.org

[11] Guoliang Xing, XiaoruiWang, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill, *Integrated coverage and connectivity configuration for energy conservation in sensor networks*, ACM Transactions on Sensor Networks, 1(1) : 36-72, 2005

[12] Tijs van Dam, Koen Langendoen, *An adaptive energy-e cient MAC protocol for wireless sensor networks*, 1st international conference on embedded networked Sensor Systems (SenSys'03), pages 171-180, New York, NY, USA, 2003

[13] S. Mahfoudh, P. Minet, *Performance evaluation of the SERENA algorithm to SchEdule RoutEr Nodes Activity in wireless ad hoc and sensor networks*, AINA 2008, IEEE 22nd International Conference on Advanced Information Networking and Applications, Ginowan, Japon, Mars 2008.

[14] Akyildiz, I. F., Melodia, T., and Chowdhury, K., *A Survey on Wireless Multimedia Sensor Networks*, Computer Networks (Elsevier) Journal, 2007

[15] OCARI presentation http://ocari.org

[16] Freescale Semiconductor, ZigBee Reference Disign (ZRD01) documentation & applications notes, 2004

[17] Jurcik, P. Koubaa, A. Alves, M. Tovar, E. Hanzalek, Z., *A Simulation Model for the IEEE 802.15.4 protocol: Delay/Throughput Evaluation of the GTS Mechanism*, MASCOTS'07: 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 24-26 Oct. 2007

[18] http://data.energizer.com/PDFs/L91ULT_EU.pdf

[19] Okazaki, S. Takahashi, and S. Higuchi, *Influence of rest timein an intermittent discharge capacity test on the resulting performance of manganese-zinc and alkaline manganese dry batteries*, Progress in Batteries & Solar Cells, 1987, v. 6, pp. 106–109.

[20] C. F. Chiasseriniand R. R. Rao, *A model for battery pulsed discharge with recovery effect*, Proc. Wireless Commun. Networking Conf., 1999, v. 2, pp. 636–639.

[21] 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

[22] C. F. Chiasseriniand R. R. Rao, *Routing protocols to maximize battery efficiency*, Proc. MILCOM, 2000, v. 1, pp. 496–500.