Sapienza University of Rome

Ph.D. program in Computer Engineering

XXIX Cycle - 2016

The Energy Problem in Resource Constrained Wireless Networks

Mario Paoli

Sapienza University of Rome

Ph.D. program in Computer Engineering

XXIX Cycle - 2016

Mario Paoli

The Energy Problem in Resource Constrained Wireless Networks

Thesis Committee	**Reviewers**
Prof. Andrea Vitaletti (Advisor)	Prof. Silvia Santini
Prof. Giorgio Grisetti (Co-Advisor)	Prof. Amy L. Murphy

Author's address:
Mario Paoli
Dipartimento di Ingegneria Informatica, Automatica, Gestionale
Sapienza Università di Roma
Via Ariosto 25, 00185 Roma, Italy
e-mail: paoli@dis.uniroma1.it
www:
http://www.dis.uniroma1.it/∼dottoratoii/students/mario-paoli

Abstract

Today Wireless Sensor Networks are part of a wider scenario involving several wireless and wired communication technology: the Internet Of Things (IoT). The IoT envisions billions of tiny embedded devices, called Smart Objects, connected in a Internet-like structure. Even if the integration of WSNs into the IoT scenario is nowadays a reality, the main bottleneck of this technology is the energy consumption of sensor nodes, which quickly deplete the limited amount of energy of available in batteries. This drawback, referred to as the *energy problem*, was addressed in a number of research papers proposing various energy optimization approaches to extend sensor nodes lifetime. However, *energy problem* is still an open issue that prevents the full exploitation of WSN technology.

This thesis investigates the *energy problem* in WSNs and introduces original solutions trying to mitigate drawbacks related to this phenomenon. Starting from solutions proposed by the research community in WSNs, we deeply investigate critical and challenging factors concerning the *energy problem* and we came out with cutting-edge low-power hardware platforms, original software energy-aware protocols and novel energy-neutral hardware/software solutions overcoming the state-of-art.

Concerning low-power hardware, we introduce the MagoNode, a new WSN mote equipped with a radio frequency (RF) front-end which enhances radio performance. We show that in real applicative contexts, the advantages introduced by the RF front-end keep packet re-trasmissions and forwards low. Furthermore, we present the ultra low-power Wake-Up Radio (WUR) system we designed and the experimental activity to validate its performance. In particular, our Wake-up Radio Receiver (WRx) features a sensitivity of -50 dBm, has a current consumption of 579nA in idle-listening and features a maximum radio range of about 19 meters. What clearly resulted from the experimental activity is that performance of the WRx is strongly affected by noise. To mitigate the impact of noise on WUR communication we implemented a Forward Error Correction (FEC) mechanism based on Hamming code. We performed several test to determine the effectiveness of the proposed solution. The outcome show that our WUR system can be employed in environment where the Bit Error Rate (BER) induced by noise is up to 10^{-2}, vice versa, when the BER induced by noise is in the order of 10^{-3} or below, it is not worth to use any Forward Error Correction (FEC) mechanism since it does not introduce any advantages compared to uncoded data.

In the context of energy-aware solutions, we present two protocols: REACTIVE and ALBA-WUR. REACTIVE is a low-power over-the-air programming (OAP) protocol we implemented to improve the energy efficiency and lower the image dissemination time of Deluge T2, a well-known OAP protocol im-

plemented in TinyOS. To prove the effectiveness of REACTIVE we compared it to Deluge exploiting a testbed made of MagoNode motes. Results of our experiments show that the image dissemination time is 7 times smaller than Deluge, while the energy consumption drops 2.6 times.
ALBA-WUR redesigns ALBA-R protocol, extending it to exploit advantages of WUR technology. We compared ALBA-R and ALBA-WUR in terms of current consumption and latency via simulations. Results show that ALBA-WUR estimated network lifetime is decades longer than that achievable by ALBA-R. Furthermore, end-to-end packet latency features by ALBA-WUR is comparable to that of ALBA-R.

While the main goal of energy optimization approaches is motes lifetime maximization, in recent years a new research branch in WSN emerged: Energy Neutrality. In contrast to lifetime maximization approach, energy neutrality foresees the perennial operation of the network. This can be achieve only making motes use the harvested energy at an appropriate rate that guarantees an everlasting lifetime. In this thesis we stress that maximizing energy efficiency of a hardware platform dedicated to WSNs is the key to reach energy neutral operation (ENO), still providing reasonable data rates and delays. To support this conjecture, we designed a new hardware platform equipped with our wake-up radio (WUR) system able to support ENO, the MagoNode++. The MagoNode++ features a energy harvester to gather energy from solar and thermoelectric sources, a ultra low power battery and power management module and our WUR system to improve the energy efficiency of wireless communications. To prove the goodness in terms of current consumption of the MagoNode++ we ran a series of experiments aimed to assess its performance. Results show that the MagoNode++ consumes only 2.8 μA in Low Power Mode with its WRx module in listening mode.

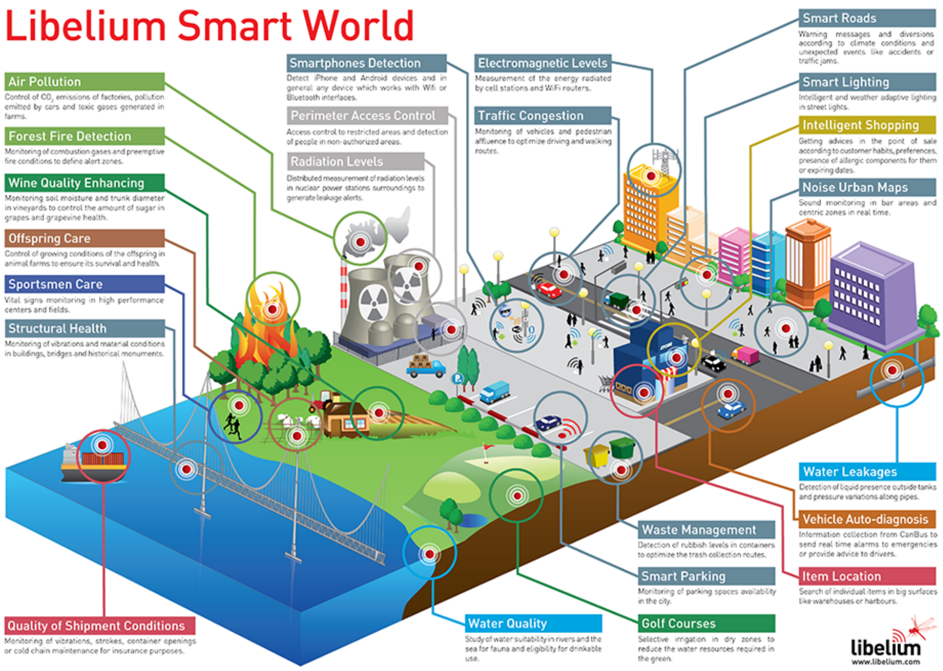
While carrying on our research work on solutions trying to mitigate the *energy problem*, we also faced a challenging application context where the employment of WSNs is considered efficient and effective: structural health monitoring (SHM). SHM deals with the early detection of damages to civil and industrial structures and is emerging as a fundamental tool to improve the safety of these critical infrastructures. In this thesis we present two real world WSNs deployment dedicated to SHM. The first concerned the monitoring of the Rome B1 Underground construction site. The goal was to monitor the structural health of a tunnel connecting two stops. The second deployment concerned the monitoring of the structural health of buildings in earthquake-stricken areas. From the experience gained during these real world deployments, we designed the Modular Monitoring System (MMS). The MMS is a new low-power platform dedicated to SHM based on the MagoNode. We validated the effectiveness of the MMS low-power design performing energy measurements during data acquisition from actual transducers.

Contents

Introduction

The idea underlying the Internet of Things (IoT) envisions billions of Smart Objects connected in a Internet-like structure [1]. A Smart Object is a tiny embedded device able to sense, compute and act [2]. In the IoT scenario Smart Objects are enhanced with communication capabilities, which allow them to interact and coordinate with each other in order to carry out more complex tasks. Nowadays network enabled Smart Objects are embedded in buildings, industrial plants, planes, cars, any kind of goods or even human beings, animals, plants, etc., making the IoT paradigm a reality [3]. This creates a new breed of applications and challenges in several fields of the mankind like home automation, healthcare, logistics, environmental monitoring, infrastructures monitoring and so on (see figure 1). Notice that, while the IoT foresees the extension of Internet Protocol Suite to all the devices, in many practical cases Smart Objects are organized in multi-hop ad hoc networks which implement solutions that were developed mostly in the Wireless Sensor Networks (WSNs) domain [5].

In the classic sense a WSN is an ad hoc network made of battery-powered electronic devices, called motes, able to coordinate their activities, in order to collect data on a phenomenon of interest such as light, heat, pressure, etc., and to wirelessly communicate in a reliable and energy efficient way the acquired information to a collection point (sink node) [6]. Compared with the wired solution, WSNs are easier to deploy and feature lower maintenance and installation costs. Thanks to their intrinsic features, WSNs are nowadays a key technology for IoT. In recent years novel standards tried to define the integration into the Internet structure of WSNs. A prominent example of such standards is 6LowPAN [7], which introduces an adaptation layer between routing and physical layers allowing the transmission of IPv6 packets through resources constrained networks. Furthermore, the Internet Engineering Task Force (IETF) Constrained RESTful Environments (CoRE) Working Group defined a Web transfer protocol called the Constrained Application Protocol (CoAP) [8] in order to introduce Web services in the IoT. CoAP is a REST-based protocol, which includes several HTTP functionalities taking into account constraints typical of WSN devices (low-power, low computa-



Libelium Smart World

Air Pollution
Control of CO_2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Forest Fire Detection
Monitoring of combustion gases and preemptive fire conditions to define alert zones.

Wine Quality Enhancing
Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

Offspring Care
Control of growing conditions of the offspring in animal farms to ensure its survival and health.

Sportsmen Care
Vital signs monitoring in high performance centers and fields.

Structural Health
Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

Smartphones Detection
Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

Perimeter Access Control
Access control to restricted areas and detection of people in non-authorized areas.

Radiation Levels
Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Electromagnetic Levels
Measurement of the energy radiated by cell stations and WiFi routers.

Traffic Congestion
Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.

Smart Roads
Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

Smart Lighting
Intelligent and weather adaptive lighting in street lights.

Intelligent Shopping
Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.

Noise Urban Maps
Sound monitoring in bar areas and centric zones in real time.

Water Leakages
Detection of liquid presence outside tanks and pressure variations along pipes.

Vehicle Auto-diagnosis
Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

Waste Management
Detection of rubbish levels in containers to optimize the trash collection routes.

Smart Parking
Monitoring of parking spaces availability in the city.

Item Location
Search of individual items in big surfaces like warehouses or harbours.

Quality of Shipment Conditions
Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.

Water Quality
Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.

Golf Courses
Selective irrigation in dry zones to reduce the water resources required in the green.

Figure 1: Internet Of Things Smart World (image courtesy of Libelium [4]).

tional capabilities). Depending on the level of integration there are three main approaches to connect a WSN to the Internet [2]. Nowadays, the most used approach consists in providing the access to the Internet solely to the sink node. As a result, the whole WSN is regarded as a single Smart Object since motes cannot directly exchange information with Internet hosts. Using this approach, motes are completely independent from the Internet and they can exploit proprietary protocols to relay the information to the sink. Another approach foresees that the sink node operates as application layer gateway able to make TCP/IP and the WSN proprietary protocols compatible. In this way, motes can be directly accessed from the Internet while the WSN is still keeping a certain level of independence from the Internet communication stack. In the last approach motes directly implements TCP/IP stack and web services interfaces and the sink node simply acts as router forwarding the packets from and to the WSN.

The Energy Problem in WSNs. Even if the integration of WSNs into the IoT scenario is nowadays a reality, the main bottleneck, that prevents the fully exploitation of this technology, is the energy consumption of motes, which quickly deplete the limited amount of energy of available in batteries.

	GOAL
Low Power Hardware	Minimize motes energy consumption. Motes are designed using state-of-the-art low energy demanding electronics components.
Energy Awareness	Maximize network lifetime. Motes run software solutions able to control in a energy efficient way the low-power hardware.
Energy Harvesting	Extend network lifetime. Motes recharge their batteries harvesting the energy available in the environment.
Energy Neutrality	Everlasting network lifetime. Motes exploit energy harvesting mechanisms and are capable to schedule tasks according to the available energy in order to avoid batteries full discharge.

Table 1: Energy Optimization Approaches.

This problem, referred to as the *energy problem*, was tackled in a number of research papers [9–24] proposing energy optimization approaches summarized in table 1. The history of energy optimization in the WSNs world begins with low-power hardware [9], continues with energy aware software/hardware solutions [10], up to energy harvesting and energy neutral systems [11, 12].

Low-power Hardware. In the last decade, thanks to the spread of portable digital devices, research in low energy electronics components became a very active field. Nowadays, a plethora of low power hardware components specifically designed for battery-powered applications are available in the market. Starting from this technological achievement, low-power hardware platforms dedicated to WSNs aim to save energy relying on low energy components. The average current consumption of most common motes with radio and CPU in the active state is about 20 mA [13]. Assuming motes to be powered by a 2000 mAh battery, it translates in a 4 day autonomy. This is not enough for highly energy-constrained systems like WSNs, which require to last for several months.

Energy awareness. Energy efficiency is a complex task to achieve since it involves the optimization of motes energy consumption exploiting also software solutions, instead of relying only on low-power hardware. Doing so, wireless sensor nodes became energy aware devices with the ability to dynamically calculate a trade-off between energy consumption, system performance and operational reliability. Moreover, the accent of energy saving needs to be placed not only on the optimization of the power consumption of a single mote, but also on maximizing the lifetime of the whole WSN. In this case, energy-aware software solutions for WSNs became distributed algorithms sharing information about energy consumption on the whole network, instead of running on a single mote. Energy aware solutions range from dynamic power management of peripherals and dynamic voltage scaling policies for microcontroller units, passing through power management of radio apparatus (energy aware MAC protocols), reaching energy aware routing protocols specifically designed for WSN [14–17]. Recently, cross-layer communication protocol were proposed to optimize energy consumption [18, 19, 25]. These protocols relay on the integration of MAC and routing layers to maximize the energy efficiency of the communication stack. Finally, the energy efficiency can be ensured through application-based energy aware protocols, which relays on techniques, such as data aggregation, able to reduce the overall network traffic. Doing so helps to decrease collisions and retrasmissions of packet, increasing the power efficiency of wireless communications [20].

Energy Harvesting. In recent years the need for WSNs able to operate for an amount of time in the order of years emerged. Fulfilling the needs of this particular application requires new energy sources to be used alongside batteries. This alternative is provided by the energy coming from renewable energy sources such as solar, wind, thermoelectric and so forth. A hardware platform able to gather energy from a renewable energy source is called energy-harvesting platform. The goal of the energy-harvesting platform is to recharge motes batteries in order to extend network lifetime. Energy harvesting is an active research branch in WSNs, indeed many energy harvesting solutions were proposed by the research community in recent years [11, 21–24].

Energy Neutrality. The naïve way of using harvested energy is to treat it as a supplement of the battery energy, trying to maximize the lifetime of motes through energy management. Anyway a different approach is possible: exploit harvested energy at a suitable rate, allowing motes to operate perennially. This mode of operation is called energy-neutral mode [12]. Using the energy-neutrality definition we can say that a mote, geared with an harvesting module, reaches energy-neutral state if it can support a certain level of workload forever. In this scope, an energy neutral mote is not just a WSN mote interfaced with an energy harvesting module, but it is an embedded system, aware of the amount of energy available in the surrounding environment and capable of predicting the quantity of energy available in the next future. These abilities allow the mote to decide which will be next task it will accomplish (i.e. sense, transmit data, go to sleep state) based on the acquired information about energy availability. To achieve this result an ad hoc hardware/software design - meant to optimize the use of the available energy, which could change over space and time and in relation to the tasks that a mote has to complete - is required.

Thesis Outline and Contributions

The thesis is organized in two parts: in part I we present the research work done trying to mitigate the *energy problem* in WSNs, while part II is dedicated to real-world WSN applications where we applied solutions introduced in part I.

Part I: The Energy Problem in WSNs

The first part of the thesis focuses on the *energy problem* in WSNs. Starting from solutions proposed by the research community in WSNs, we deeply investigate critical and challenging factors concerning the *energy problem* and we came out with cutting-edge hardware platforms and original software protocols overcoming the state-of-art. In Chapter 1 and 2 we focused our attention on increasing the energy efficiency of motes and of communication protocols dedicated to WSNs, while chapter 3 is dedicated to our research work in the relatively new field of energy neutrality. It is important to point out that to carry on a research work in energy neutrality the energy harvesting field need to be previously investigated. In this thesis we explored the energy harvesting topic only in the scope of energy neutrality. Figure 2 summarizes contributions of this thesis to the *energy problem* in Wireless Sensor Networks.

Chapter 1 - Low-Power Hardware. Chapter 1 is dedicated to low-power hardware platform we developed in order to increase the energy efficiency of sensor nodes. In the specific, section 1.1 describes in details the hardware and software architecture of the MagoNode, a new low-power mote operating in the 2.4 Ghz ISM band we designed and developed. The MagoNode is based on a highly efficient RF front-end which enhances radio performance, without giving up energy efficiency. Indeed, the power consumption of our mote is lower than other existing sensor nodes featuring a RF front-end and it is comparable to most commonly known motes (see section 1.1.5 for details). In section 1.2 we introduce the research work done in the scope of Radio Wake-Up technology. A wake-up radio receiver (WRx) is an almost passive hardware

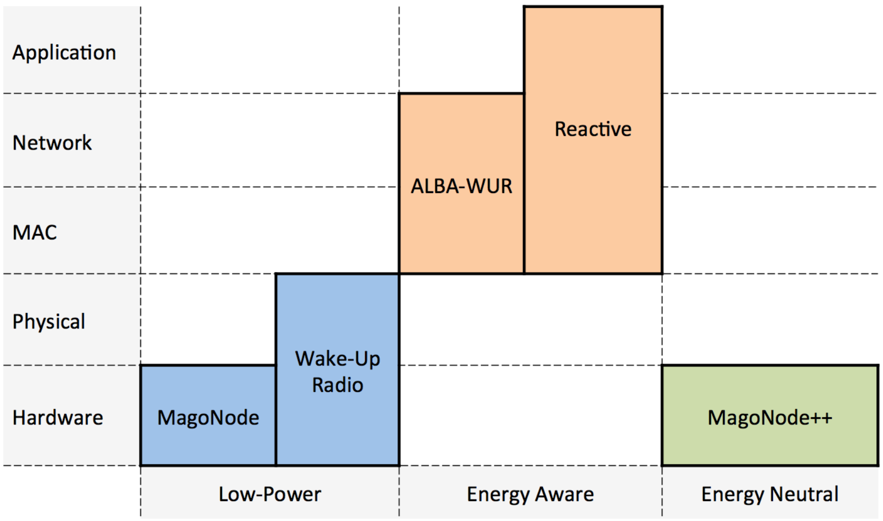


Figure 2: Thesis Contributions.

module which continuously listen to the channel and awakes the mote main transceiver only when a message is detected. Doing so, the WRx increases the energy efficiency preventing mote transceiver idle listening, which is considered the primary source of energy waste in WSNs [26]. The wake-up radio (WUR) system we designed is based on OOK modulation and features an ultra low-power WRx and a wake-up transmitter (WTx) able to send wake-up messages with an output power up to +12 dBm. Our WRx requires an amount of power in the order of nano-watts to operate in idle listening state. This value is about four orders of magnitude lower than standard transceivers, which are characterized by milli-watts consumption. Our wake-up radio system was implemented on a Printed Circuit Board (PCB) which can be easily interfaced to the MagoNode, enhancing mote features and energy efficiency. The superior performance of our WRx in terms of power consumption affects the radio range and the robustness of wireless communications. Indeed, our WRx is sensible to RF interference, which translates into noise perceived by the receiver, causing spurious awakenings of the mote and errors in received wake-up messages. To overcome these issues we developed hardware solutions able to limit the spurious awakenings and with software solutions able to increase the robustness of communication.

Chapter 2 - Energy Awareness. Even if low power consumption of motes hardware is a crucial aspect to maximize the energy efficiency, it is not enough to meet lifetime requirements of WSN applications. Therefore, energy awareness need to be introduced in WSNs software architecture. As already pointed out, the energy awareness concept can be implemented at different levels: starting from the application-layer, until reaching the physical layer. In chapter 2 we introduce the research work done improving the energy efficiency of protocols dedicated to WSNs introducing energy-aware solutions. In the specific, in section 2.1 we analyze the interaction between two communication protocols implemented in TinyOS 2.x [27]: Deluge T2, an over-the-air programming (OAP) protocol, and the Low Power Listening (LPL) implementation provided with the standard Medium Access Control layer, BoX-MAC [28]. We show how the characteristics of the two layers deeply diverge, leading to a sensible performance degradation. To overcome this issue, we implemented REACTIVE, a energy aware OAP protocol, which is able to dynamically disable and re-enable LPL in order to boost Deluge performance. REACTIVE is able to increase the performance of Deluge by a factor of 2.6 in terms of energy efficiency and 7 in terms of dissemination time. Furthermore, in section 2.2 we present the ALBA-WUR protocol, a cross-layer solution for data gathering in WSNs that redesigns a leading routing protocol, ALBA-R [29], extending it to exploit wake-up radio benefits. With ALBA-WUR we show that the complex relay selection policies used by ALBA-R can be naturally expressed by one or more semantic addresses signifying the current status of a node, which the node itself can locally compute and take over time. A sender can therefore selectively wake-up only those neighbors whose semantic addresses match the policy, thus engaging in communications only nodes whose current status makes them the best relays. We compare the performance of ALBA-WUR and ALBA-R via simulations. Performance results in WSN scenarios with increasing traffic show impressive performance gains of ALBA-WUR over ALBA-R (see section 2.2.3 for more details).

Chapter 3 - Energy Neutrality. In recent years the combination of low-power design, energy harvesting and ultra-low-power wake-up radios is paving the way for energy neutral operation (ENO) of WSNs. As already pointed out, a node operates in ENO mode if the energy it consumes over a given time period is no greater than the amount of battery charge plus the energy harvested during the same time period. Depending on the considered scenario, achieving ENO may be challenging. For example, indoor energy sources such as ambient light in illuminated rooms can provide power only in the order of tens of μW per cm^2. To support energy-neutral operation in such scenarios, the energy consumption of the node must be reduced as much as possible. The use of a wake-up receiver (WRx) drastically increases the energy efficiency of wireless communication by virtually eliminating idle listening, which is one of the main sources of energy waste in WSNs. In section 3.1, we present the MagoNode++, a novel WSN platform supporting radio-triggered wake-ups for energy-neutral applications. The MagoNode++ features an energy harvesting subsystem composed by a light or thermoelectric harvester, a battery manager and a power manager module and integrates our wake-up receiver to boost the efficiency of radio communications. Experimental results show that the MagoNode++ consumes only 2.8 μA with the WRx in idle listening and the rest of the platform in sleep state, outperforming existing prototypes featuring a WUR system such as the one presented in [30], with a reduction of almost 50% in energy consumption. Low power consumption and energy-harvesting and radio-triggering capabilities of the MagoNode++ make it suitable for challenging application scenarios that requires high energy efficiency and self-sustainable operations.

Part II: WSN For Structural Health Monitoring

In this part of the thesis we describe how we applied the solutions presented in part I to structural health monitoring (SHM). SHM deals with the early detection of damages to civil and industrial structures, such as roads, bridges, canals, buildings and aero-space vehicles, etc., and is emerging as a fundamental tool to improve the safety and maintainability of these critical infrastructures. SHM requires a variety of sensing technologies together with an embedded system able to capture, log, and analyze data in order to prevent collapses and breaks, avoiding permanent damages to structures, thus simplifying and improving the effectiveness of their maintenance. Depending on application scenarios, many different types of sensors are required, including pressure sensors, vibrating-wire strain gauges, inclinometers, crackmeters, etc.

Nowadays, most of the commercial SHM systems are wired. However, the deployment of such systems in a wide area or in a harsh environment, poses both economical and practical problems. For this reason, WSNs have been proposed as ad hoc network infrastructures to support SHM applications. SHM supported by WSN is an active and well-established research field [31–33] and some wireless SHM systems are now entering the market.

Chapter 4 - Underground Construction Site Monitoring In chapter 4 we present the WSN we designed to monitor the construction site of Rome B1 underground. The goal of the testbed was to gather structural health data during the excavation of a 700 m tunnel connecting two stops. The WSN we deployed to achieve this result was Composed byof 32 wireless nodes, 49 transducers and a sink node connected to the mains. The network operated for whole digging period, which began in November 2011 and ended in February 2012. To relay data in an energy efficient way we used the DISSense, an ultra-low power protocol specifically designed for periodic data collection [25]. DISSense features very low radio duty cycle, enabling the WSN motes to operate for years relying only on energy stored in batteries. Experimental data collected during the testbed showed that the average Data Delivery Ratio (DDR) (i.e. the percentage of data packets transmitted by motes actually received by the sink) was higher than 95% and that the average radio duty cycle of motes was 0,22%.

Chapter 5 - Earthquake-stricken Areas Monitoring In chapter 5 we introduce our work concerning SHM in earthquake-stricken areas. In the specific, we design and realized a network featuring sensor nodes able to monitor damaged buildings and to acquire information about the quality of the air, which could be impaired due to gas leakages. All the work done was carried out in the context of the research project "TETRis - TETRA Innovative Open Source Services" [34]. The main goal of TETRis was to extend the public safety services and applications running on top of the Terrestrial Trunked Radio (TETRA) [35] communication system with real-time data collected by WSNs deployed on field. In the scope of TETRis project we designed and realized a WSN able to gather data in earthquake-stricken areas. In particular, our WSN is able to monitor damaged buildings and to acquire information about the quality of the air, which could be impaired due to gas leakages. As final demonstrators of the project, we deployed an hybrid indoor/outdoor testbed consistent with the reference TETRis scenario. The results of the experimental activity confirmed that WSNs can be effectively used to support the management of critical situations in earthquake-stricken areas.

Modular Monitoring System From the experience gained during on-field experiences, we decided to design a new wireless, low-power, scalable hardware architecture dedicated to SHM: the Modular Monitoring System (MMS) (see section 6). A key feature of the MMS is the modularity which allows easy customization of the platform depending on the number and the type of sensors required by the specific SHM scenario. The MMS supports mesh network topology and offers excellent coverage and reliability, taking advantage of WSN technology. Thanks to its peculiar characteristics, the MMS overcomes commercial state-of-the-art WSN solutions for SHM, which do not offer enough flexibility to fulfill requirements of SHM scenarios in a both cost-effective and efficient way.

Publications

Part of these thesis was published in the following journal articles and conference and workshop proceedings:

1 *MagoNode: Advantages of RF Front-ends in Wireless Sensor Networks.*

Mario Paoli, Antonio Lo Russo, Ugo Maria Colesanti, Andrea Vitaletti.

REALWSN 2013.

2 *Structural health monitoring in an underground construction site: the roman experience (Poster).*

Ugo Maria Colesanti, Antonio Lo Russo, Mario Paoli, Chiara Petrioli, Andrea Vitaletti.

ACM SenSys 2013.

3 *Introducing the MagoNode platform (Demo).*

Ugo Maria Colesanti, Antonio Lo Russo, Mario Paoli, Chiara Petrioli, Andrea Vitaletti.

ACM SenSys 2013.

4 *REACTIVE: A peaceful coexistence between deluge and Low Power Listening.*

Andrea di Cagno, Mario Paoli, Ugo Maria Colesanti, Andrea Vitaletti.

WINMEE - WiOpt 2014.

5 *Extending TETRA with Wireless Sensor Networks.*

Mario Paoli, Andrea Di Cagno, Ugo Maria Colesanti, Andrea Vitaletti, Simona Citrigno, Domenico Saccà.

International Journal of Intelligent Engineering Informatics 3.2-3 (2015): 225-243.

6 *Beyond Duty Cycling: Wake-up Radio with Selective awakenings for Long-lived Wireless Sensing Systems.*

Dora Spenza , Michele Magno, Stefano Basagni, Luca Benini, Mario Paoli , Chiara Petrioli.

INFOCOM 2015.

7 *Wireless Sensor Networks in Structural Health Monitoring: a Modular Approach.*

Fabio Angeletti, Mario Paoli, Ugo Maria Colesanti, Andrea Vitaletti.

SENSORCOMM 2015.

8 *Further steps into Wireless Sensor Networks in Structural Health Monitoring.*

Fabio Angeletti, Mario Paoli, Ugo Maria Colesanti, Andrea Vitaletti.

Sensors & Transducers Journal, volume 194.

9 *MagoNode++: a Wake-Up-Radio-Enabled Wireless Sensor Mote for Energy-Neutral Applications (Poster).*

Mario Paoli, Dora Spenza, Chiara Petrioli, Michele Magno, Luca Benini.

IPSN 2016.

Part I

The Energy Problem in Wireless Sensor Networks

The main bottleneck of WSNs, which prevents the fully exploitation of such technology, is the energy consumption of motes which quickly deplete the limited amount of energy of available in batteries, shrinking network lifetime and operation. This problem, referred to the *energy problem*, were tackled in a number of research papers [9–24] proposing energy optimization approaches ranging from low power hardware, passing through energy aware protocols and energy harvesting platform, reaching new energy exploitation approaches. In this part we will introduce original solutions to the *energy problem* we designed and implemented. In particular, chapter 1 is dedicated to low-power hardware platform we developed in order to increase the energy efficiency of sensor nodes. Even if low power consumption of motes hardware is a crucial aspect to maximize the energy efficiency, it is not enough to meet lifetime requirements of applications. Therefore, energy awareness need to be introduced in WSNs software architecture. In chapter 2 we will present the research work done improving the energy efficiency of protocols dedicated to WSNs introducing energy-aware solutions. In recent years the combination of low-power design, energy harvesting and ultra-low-power wake-up radios (WURs) is paving the way for energy neutral operation (ENO) of WSNs. A node operates in ENO mode if the energy it consumes over a given time period is no greater than the amount of battery charge plus the energy harvested during the same time period. In chapter 3 we will introduce our energy neutral hardware platform which take advantage of WUR technology to support network energy-neutral operation.

Chapter 1

Low-Power Hardware Platforms

Minimizing the energy consumption is a primary requirement to extend network lifetime. Indeed, despite the distributed nature of Wireless Sensor Networks (WSNs), the exhaustion of motes batteries could lead to network partitioning and connectivity problems which may cause network malfunctioning. Nowadays, a great variety of low-power hardware components designed for battery-powered embedded systems are available in the market. Taking advantage of these components while designing a sensor node hardware architecture is mandatory and is the first step toward the achievement of energy efficient WSNs hardware platforms.

In this chapter we present our research work trying to extend motes lifetime through the design of innovative low-power hardware platforms. In particular, we detail the realization of the MagoNode (see section 1.1), a new low-power mote featuring a radio front-end enhancing radio performance, without giving up energy efficiency. As we will show in section 1.1.6, the advantages introduced by the radio front-end, such as improved communication reliability and extended radio ranges, allow the MagoNode to limit the energy consumption by shrinking the number of retransmitted packets and minimizing the hop count to reach a destination. Moreover, the power consumption of our mote is lower than other existing sensor nodes featuring a RF front-end and it is comparable to most commonly known motes. Furthermore, in sections 1.2, 1.3, 1.4, 1.5, 1.6 we will introduce the research work and the experimental activities done in the scope of wake-up radio (WUR) technology. A wake-up radio receiver (WRx) is an almost passive hardware module featuring current consumption in the order of hundreds of nA, almost two order of magnitudes less than state-of-the-art transceivers. The main duty of a WRx module is to continuously listen to the channel and wake the mote when traffic over the channel is overheard. In this way it is possible to limit the idle listening time of the main transceiver, which is considered the primary source of energy waste in WSNs [26]. Indeed, the main transceiver can stay off until a message has to be actually received since it will be awaken by the WRx when traffic is detected over the channel.

1.1 MagoNode

Most of the modern hardware platforms dedicated to WSNs use IEEE 802.15.4 [36] compliant transceivers operating in the industrial, scientific and medical (ISM) 2.4 GHz band [37, 38]. Advantages are the world-wide availability of the 2.4 GHz band, higher data rates with respect to other ISM bands (433 MHz, 868/915 MHz), almost no regulatory restrictions related to the duty cycle [39,40] and compliance to ZigBee, that is a widespread standard solution for remote monitoring and control applications [41]. Unfortunately, those advantages come at the expense of a lower transceiver sensibility and higher propagation losses due to the higher transmission frequency which translates in a shorter radio range, especially for indoor applications.

In several practical scenarios, the limited radio range of IEEE 802.15.4 transceivers requires the presence of many additional relay nodes to guarantee network connectivity. Common assumptions in theoretical works on WSNs are that the density of the network can be arbitrarily high to guarantee the robustness of the network and to improve the quality of collected data. However, in many practical cases those assumptions are questionable. In our experience in monitoring critical infrastructures, a limited number of very accurate and costly transducers need to be deployed in given positions potentially far from each other. This implies several relaying nodes to be placed in order to guarantee a reliable and robust network. However, a high number of relay nodes implies higher hardware and deployment costs as well as an increase in energy costs due to more complex coordination and management activities. As an example, in our WSN test-bed of the Roman B1 line underground construction site, a 700 meters tunnel has been monitored by our network for months (from November 2011 until June 2012) [42]. To guarantee wireless connectivity and an acceptable link quality, nodes were about 40 to 50 meters apart. However, experts required the monitoring of only 4 sections of the tunnel spaced from 100 to 400 meters from each other. Hence, at the end of the deployment, the tunnel was monitored by 32 nodes: one acting as sink, 23 as relay nodes and only 8 as nodes interfaced with transducers. The presence of many relay nodes increased network deployment cost, latency to route packets toward the sink and an overhead in terms of consumption due to management communications. To overcome these issues, some recent hardware solutions for WSNs have started to propose built-in radio frequency (RF) front-ends [43, 44]. An RF front-end is a combination of a power amplifier (PA) and a low noise amplifier (LNA) that enhances the transmission and the reception of the transceiver, at the expense of an energy cost overhead. However, the always increasing efficiency of transceivers combined with the constantly lowering power consumption of RF front-ends make this kind of solution increasingly convenient.

In this section we present the MagoNode, a new WSN low-power platform developed at Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti" of Sapienza University of Rome in cooperation with the spin-off WSense s.r.l. [45] This platform takes advantage of the hardware improvements discussed earlier, and provides outstanding radio performance, still keeping the advantages of standard 2.4 Ghz transceivers. In the remainder of this section we briefly describe commercial RF front-ends features and the most recent nodes equipped with such devices. Then, we introduce the MagoNode platform with its extension boards and compare it with other amplified motes. We show that our platform is more flexible and has better performance in terms of power consumption than existing motes equipped with a RF front-end and, at the same time, it is still competitive compared to traditional unamplified nodes. Finally, in the last section we compare the MagoNode platform and the IRIS node [46] while running an indoor test-bed, to evaluate the advantages of an RF front-end approach.

1.1.1 RF Front-ends and Platforms

As mentioned in the previous section, an RF front-end embeds a low noise amplifier (LNA) to improve the receiver sensitivity and a power amplifier (PA) to increase the output power. In addition to several research papers focused on low-power 2.4 GHz RF front-ends for WSNs [47, 48], a wide range of devices are now available at most common RF hardware manufacturers. We took care of RF front-ends suitable for WSN applications, relying on the following parameters:

- **Supply voltage**: most wireless nodes operates in the 1.8 - 3.6 V range in order to be powered by 2x AA 1.5V batteries, hence, we expect a front-end to be as close as possible to this range.

- **Reception (RX) Gain** and **RX Current**: keeping in mind that idle listening is the most common source of energy waste in a WSN [49], the RX current overhead of the front-end must be as low as possible, still guaranteeing a good RX gain.

- **Transmission (TX) Gain** and **TX Current**: the TX power output is regulated by local organizations such as European Telecommunications Standards Institute (ETSI) in Europe and Federal Communications Commission (FCC) in North America. In particular, ETSI limits the maximum output power to +20 dBm [39, par. 4.3.2.1] but the actual output power is limited by the power spectral density specification in [39, par. 4.3.2.2] which reduces the limit to approximately +10 dBm [50]. Similarly, the FCC has a maximum output power limit of

+30 dBm [40], however, the stricter radiation emission limit, difficult to meet without expensive filtering, reduces the actual power to about +18 dBm [50]. It is important to note that the TX current overhead is typically much higher than the RX one but, at the same time, in WSNs the time spent in TX and RX is much lower than the time spent in idle listening [49].

- **Noise figure**: each RF front-end amplifies the RF signal together with RF noise. The lowest is the noise figure, the better the transceiver behaves.

Table 1.1: Front-ends characteristics (part 1).

Brand	Model	Current RX (mA)	Gain RX (dB)	Current TX (mA) @10dBm	Current TX (mA) @20dBm	Noise Fig. (dB)
Skyworks	SE2431L	5	12.5	50	115	2
Skyworks	SE2438T	5.5	10.5	20	N/A	3.5
Skyworks	SKY65344	7	10	N/A	105	2.2
Skyworks	SKY65352	7	10	72	110	2
RFMD	RF6555	8	11	60	90	3
RFMD	RF6575	8	13	90	170	2.5
Atmel	T7024	8	16	50	110	2.1
TI	CC2590	3.4	11.4	17	N/A	4.6
TI	CC2591	3.4	11	50	105	4.6
RFAxis	RFX2401C	10	12	N/A	100	2.4
RFAxis	RFX2411	8	12	N/A	110	2.5

In tables 1.1 and 1.2 we have compared several RF front-ends products available on the market based on the aforementioned parameters. The values indicated in these tables have been taken or derived from products data-sheets. It is clear that there is no evident winning solution: the best RX performance is given by CC2590 and T7024, but the former has a high noise figure and is limited to a maximum power output (P_{out}) of +14 dBm in TX while the latter, despite the good current/gain ratio, has a high RX current consumption. The best TX performance is given by RF6555 which is weak in RX due to a high current drain. Skyworks SE2431L seems to represent a good balance between TX and RX modes and natively supports antenna diversity option.

However, using an RF front-end tailored for 20 dBm power output becomes counterproductive when the P_{out} is limited to 10 dBm. In fact, using a front-end with lower maximum P_{out} like the CC2590 or SE2438T allows to reach current consumption at 10dBm that is less than half with respect to the other solutions. Based on the previous observation, it is possible to design efficient solutions fitted for different market areas and regulations. For the same reason, Texas Instruments developed CC2590 and CC2591 as interchangeable devices that share the same footprint and components.

Table 1.2: Front-ends characteristics (part 2).

Brand	Model	Max P_{out} (dBm)	Voltage (V)	Comments
Skyworks	SE2431L	23	2 - 3.6	Antenna div.
Skyworks	SE2438T	16	2 - 3.6	Antenna div.
Skyworks	SKY65344	20	2.7 - 3.6	
Skyworks	SKY65352	20	2.7 - 3.6	
RFMD	RF6555	20	2 - 3.6	Antenna div.
RFMD	RF6575	22	3 - 3.6	Antenna div.
Atmel	T7024	23	2.7 - 4.6	
TI	CC2590	14	2 - 3.6	footprint CC2591
TI	CC2591	22	2 - 3.6	footprint CC2590
RFAxis	RFX2401C	22	2 - 3.6	
RFAxis	RFX2411	21	2 - 3.6	

1.1.2 Competitors: Motes with an RF Front-end

In this subsection we briefly introduce hardware platform dedicated to Wireless Sensor Networks featuring RF front-ends.

The AdvanticSYS CM3300 node is equipped with a separated PA and LNA solution, the SiGe PA2423L and Sirenza SGL-0622Z respectively [51]. The CM3300 features 120mA current consumption for transmission at 20dBm and 30 mA current consumption in RX giving 23 dB gain and is TelosB compatible, hence, it supports TinyOS [52] and Contiki [53] operating systems. However, as opposed to other existing solutions, the CM3300 is designed to operate from mains since the 5mA sleep current does not allow it to be battery-powered for long lasting deployments.

The Atmel ZigBit Amp ATZB-A24-UFL (ZigBit) [43] is an OEM module handling Atmel 1281v MCU, AT86RF230 transceiver and a Skyworks (formerly SiGe) SE2431L RF front-end. The ZigBit accepts 3 to 3.6 V input voltage, it features +20 dBm output power and 12.5 dB of RX Gain.

The Dresden Elektronic deRFmega128-22M12 (deRFmega) is the most recent wireless OEM module available on the market [44]. It is based on Atmel

Atmega128RFA1 MCU/Transceiver System-on-chip bundle and the Skyworks SE2431L front-end. The device accepts from 2 V to 3.6 V in input and has the following characteristics (derived from the user manual): 18.5 mA RX power consumption with 12.5 dB gain, 143 mA for TX@20dBm and 59 mA for TX@10dBm. The deRFMega also supports Contiki and TinyOS operating systems.

Looking for a greater energy efficiency, we decided to design the MagoNode. Indeed, as we will show in section 1.1.5 our mote features lower power consumption than above cited platform.

1.1.3 MagoNode Hardware Platform

While the integration with an RF front-end represented the predominant part of the platform development, several other aspects, closer to commonly known wireless sensor nodes requirements, have been taken into account during the MagoNode design.

1.1.3.1 Platform Design

Controller selection We looked toward a cheap, low-power microcontroller and transceiver bundle, possibly with software support for existing WSN operating systems (TinyOS, Contiki) and with all commonly used digital and analog interfaces. Atmel Atmega128RFA1 (RFA1) met all our requirements being a 8-bit, 16 MHz system-on-chip (SoC) 802.15.4 compliant with 128KB of read-only memory (ROM), 16 KB of random-access memory (RAM) and with outstanding transceiver performance: 103.5dB link budget, a current consumption of 12.5 mA in RX and 14.5 mA in TX@3.5dBm. The SoC characteristic of the RFA1 chip reduces both size and bill of materials and keeps the overall platform costs very low.

The RFA1 supports Atmel BitCloud ZigBee stack and is natively supported by TinyOS since release 2.1.2 thanks to the UCMote mini platform [38]. Also Contiki OS supports the RFA1.

RF front-end. We based our RF front-end selection on two fundamentals observations: first of all, we expect idle listening to represent the predominant energy cost on a wireless node, hence, it becomes important to pay as little RX overhead as possible when adding an RF front-end. Secondly, as already mentioned in section 1.1.1, local regulations considerably limit the maximum output power. Thus, it makes nonsense to use a +20 dBm designed front-end in Europe while it would be limiting to use a +10 dBm designed front-end in North America. At the same time, it is very unlikely that the same hardware used in one geographical area will move in a completely different region.

Based on these observations and having a look back to table 1.1, we observe how CC2590 and CC2591 represent an optimal solution for our requirements. In particular, both front-ends have the lowest RX current consumption still guaranteeing good RX gains and, most importantly, they both share the same footprint, making them interchangeable over the same PCB design.

PCB design. The MagoNode platform is designed as a 4-layer original equipment manufacturer (OEM) board, i.e., a castellated printed circuit board (PCB) solderable on bigger application-specific boards. The reason of such choice with respect to more academic-like solutions such as TelosB or MicaZ, resides in a higher flexibility in board design for end-users applications. The MagoNode OEM board is 35 mm long and 20 mm large (figure 1.1). It already holds the radio frequency (RF) filtering section designed following [54] and a U.FL connector that allows the connection of an external antenna plug. The board has 39 castellated pins featuring most of RFA1 capabilities, including 2 universal asynchronous receiver/transmitter (UART) interfaces, 1 serial peripheral interface (SPI), 1 inter-integrated circuit I^2C, more than 20 general purpose input/output (GPIO) pins and 4 10-bit analog-to-digital convertes (ADC) channels. One pin can be enabled as antenna output by moving a 0-Ohm resistor. Finally, the board footprint has been designed to enable antenna-diversity in future releases.

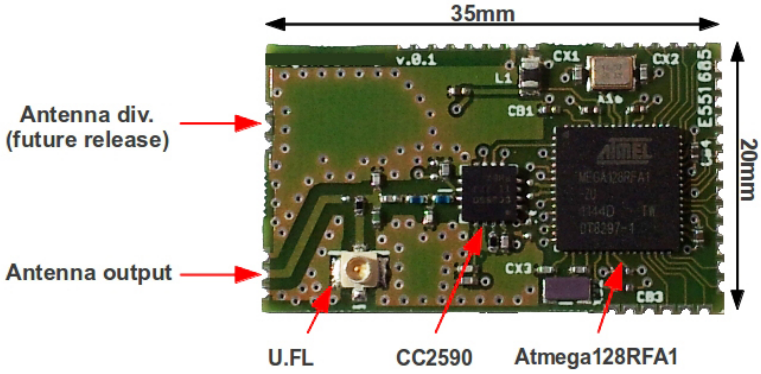


Figure 1.1: The MagoNode OEM platform.

Low power mode. The lifetime of a mote is divided between periods of activity and periods of inactivity. During periods of activity motes perform typical WSN operations such as receiving a packet, sending a packet, sensing the quantity of interest and so on. On the other side, during periods of inactivity, sensor nodes switch to low power mode turning off all hardware peripherals and putting the microcontroller in the lowest power consumption state in which RAM memory is retained and an hardware timer, which schedules next period of activity, is active. In typical low data rate WSN applications the amount of time spent by motes in active states it is just a small portion compared to the amount of time they spend in the sleep state waiting to perform a task. For such reason, in order to maximize mote lifetime, it is mandatory to keep low power mode consumption of a hardware platform dedicated to WSN as low as possible. While designing the MagoNode we took particular care to this crucial aspect. Indeed, our platform consumes only 300 nA in the sleep state. This current is necessary to keep on a 32 kHz oscillator feeding a timer which schedules next period of activity and a module of the microcontroller able to catch external interrupts. In such way, the MagoNode can be awaken after a certain amount of time defined during a period of activity (e.g. sampling time) or when an asynchronous events happens (e.g. an external peripheral wants to awake the mote).

1.1.3.2 Boards

As as a first step, we integrated the MagoNode in two different boards (figure 1.2). The first one, named MNA-Board, is a general purpose board featuring a 2x AA battery holder, a power switch, three debug leds, an RP-SMA connector, a 51 pin Hirose expansion connector and, optionally, a 2MB flash chip. The MNA-Board is 32 mm large and 55 mm long and, as almost all commonly known wireless sensor nodes, allows quick prototyping and debugging. The second board is an application-specific sensor-board which acts as an interface for 4 - 20 mA Current-Loop sensors. The 55 x 55 mm board accepts 5 - 30 V input voltage, provides a digitally controlled power switch for the current loop sensor and has three 24-bit ADC channels with precision resistors for current measurements. The board is housed in an IP56 box and uses an external antenna plugged to the U.FL connector of the MagoNode.

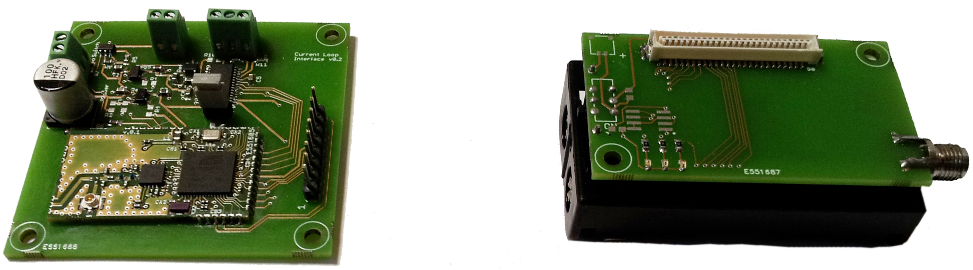


Figure 1.2: The current-loop interface (left) and the MNA-board (right).

1.1.4 MagoNode Software Platform

One of the reason why we equipped the MagoNode with the RFA1 lies in the fact that this microcontroller benefits of large software support. In particular it is supported by two of the most popular open-source operating system dedicated to WSNs and embedded systems: TinyOS and Contiki [52, 53]. Right now all our efforts creating a stable and robust software platform for the MagoNode have been done using TinyOS, anyway, we do not rule out to do a porting on Contiki OS in the next future.

TinyOS officially supports the RFA1 since the version 2.1.2. Besides software modules and drivers which allow easy access to RFA1 features such as GPIO, serial interfaces like I^2C, SPI, UART, ADC, timers and interrupts, TinyOS offers also widely tested drivers for the radio apparatus of the RFA1. Moreover, TinyOS provides a robust layer two radio stack, called RFXLink, which is compliant with the 802.15.4 standard. The RFXLink implements Low Power Listening (LPL) [55], a MAC level mechanism which allow to duty cycle the radio activities in order to save energy, and features mechanisms to enable collision avoidance to mitigate the number of retransmitted packets. In the specific, the RFXLink implements two collision avoidance algorithms: the first one is based on random backoff time which nodes have to wait before transmitting a packet. The other is a Time Division Multiple Access (TDMA) mechanism: time is divided in slots which are assigned to node of the network. Each node is allowed to transmit only in its own time slot. Furthermore, the RFXLink implements a flexible retransmission module which allows packets to be automatically retransmitted a number of time that is configurable at coding time. Least but not last, the RFXLink has a traffic monitor module which gives statistics about number of packets transmitted and received and about the amount of time the radio apparatus spent in transmission, reception and idle listening. On top of the RFXLink, which provides robust and reliable layer two connectivity, it is possible to put one of the layer three routing protocol implemented by TinyOS. Among these protocols the Collection Tree Protocol (CTP) [56, 57] is a de-facto standard for low-power data collection in WSN.

A mote featuring a software platform with a robust communication stack and reliable drivers to control application specific peripherals is mandatory in both real world scenarios and evaluation testbeds.

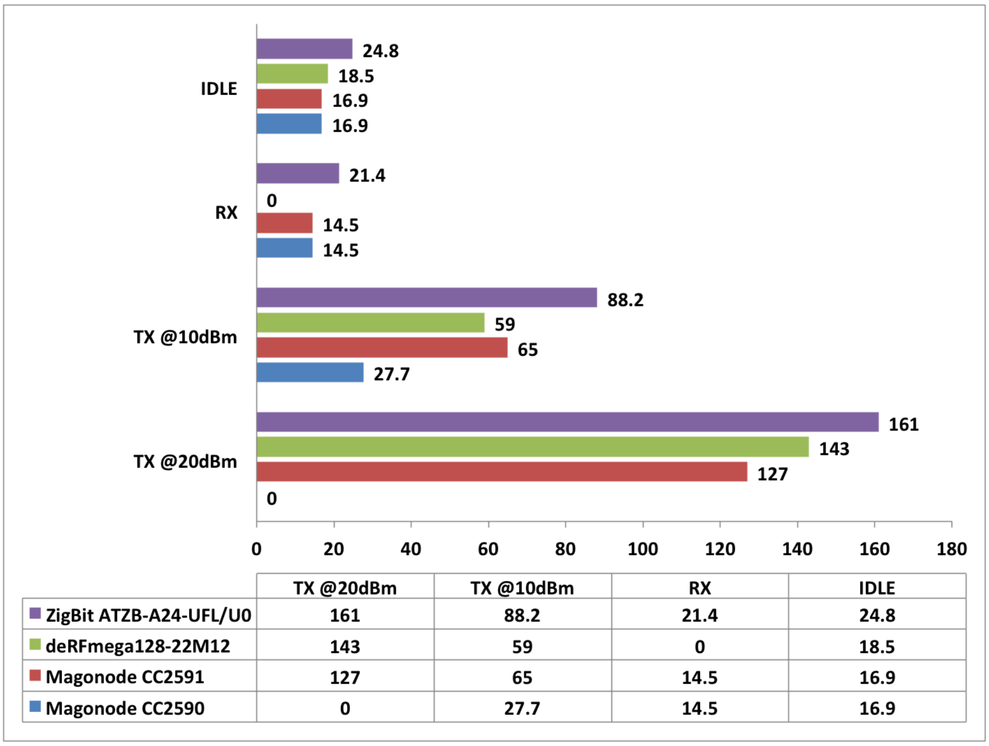
1.1.5 Radio Current Consumption: Performance Comparison

To evaluate the performance of the MagoNode platform, in terms of current consumption, we have run two sets of tests. On the one side we compared the European (CC2590) and North American (CC2591) versions of the MagoNode platform with respect to similar state-of-art amplified wireless nodes mentioned in section 1.1.2, i.e., ZigBit and deRFMega. On the other side, we compared the CC2590 version of the MagoNode with respect to common unamplified motes like IRIS, TelosB and the Atmel Atmega128RFA1-EK1 which shares the same MCU/Transceiver bundle with the MagoNode but has not an RF front-end. Actually, we want to demonstrate not only the competitiveness of our solution compared to other amplified devices, but also that the energy overhead introduced by the MagoNode is low enough to get compared with unamplified motes.

We programmed each device with the RadioCountsToLeds application available in the official TinyOS 2.1.2 distribution [52]. With a Rigol DM3068 precision multimeter we sampled the current consumption with a period of 100μsec. Finally, we calculated the average current consumption of each platform in reception (RX), transmission (TX) and idle listening (IDLE).

Figure 1.3 shows the current consumption of amplified nodes with the radio in TX@20dBm, in TX@10dBm, in RX and in IDLE. At highest output power, performance of the MagoNode CC2591 overcomes that of deRFmega and that of Zigbit by 12% and 21% respectively. At 10 dBm the MagoNode CC2590, which is optimized for this power setting, has a current consumption that is less than half with respect to other considered motes. This supports the choice we made to equip the MagoNode with two interchangeable front-ends that provide the best performance at 10 dBm and 20 dBm respectively. In addition, the IDLE performance of CC2590 and CC2591 overcome the one of deRFMega by 9% and that of ZigBit by 32%. Similar results arise in RX between MagoNode and ZigBit platforms, while it was not possible to compare the deRFMega since the actual RX consumption was not reported in the datasheet.

Figure 1.4 shows the current consumption of unamplified nodes and that of the MagoNode CC2590 with the radio in TX, in RX and in IDLE. We observe how the MagoNode CC2590 has a sensibly higher current consumption in TX than those of IRIS@3dBm (29%), TelosB@0dBm (45%) and EK1@3,5dBm (90%). However, both RX and IDLE consumption of the CC2590 are lower than IRIS and TelosB by 19% and 35% respectively. Of course, being the EK1 similar to a non-amplified version of the MagoNode, even in these two radio states it drains less current by 28% in IDLE and 12% in RX. Anyway, as we will show in the next section, the employment of the MagoNode in a real multi-hop scenario, thanks to the extended radio range and enhanced link quality guaranteed by the RF front-end, results in a reduction of the network traffic due to a lower number of re-transmissions and forwards. This implies that the measured higher current consumption of the MagoNode is mitigated by an overall lower number of expected transmissions.



	TX @20dBm	TX @10dBm	RX	IDLE
🟪 ZigBit ATZB-A24-UFL/U0	161	88.2	21.4	24.8
🟩 deRFmega128-22M12	143	59	0	18.5
🟥 Magonode CC2591	127	65	14.5	16.9
🟦 Magonode CC2590	0	27.7	14.5	16.9

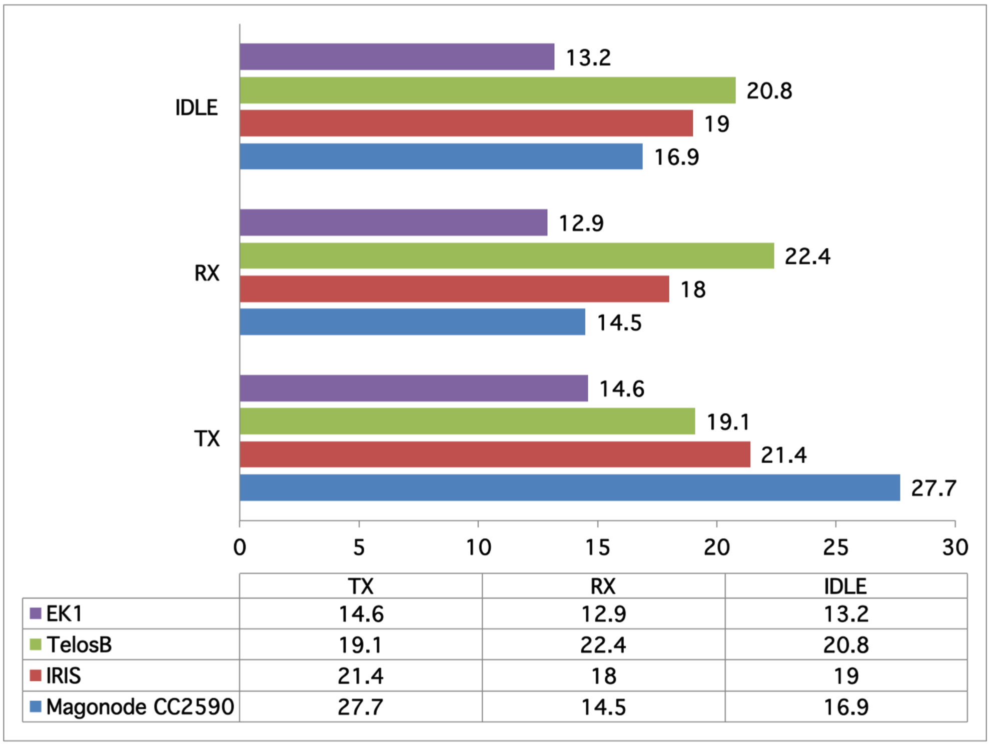
Figure 1.3: Current consumption comparison between the MagoNode and other amplified motes. The CC2590 has a maximum $P_{out} = 14dBm$ and deRFMega datasheet does not provide the RX current consumption.

1.1.6 Testbed

In the previous section we observed how the MagoNode platform outperforms other amplified solutions. At the same time we also showed how the performance of the CC2590 version of the MagoNode are close to the ones of unamplified nodes. These results lead us to think that, in a real multi-hop deployment, where the actual energy consumption is driven by re-transmissions and hop count, the benefits introduced by the front-end we chose for the MagoNode platform (CC2590) can further reduce the energy overhead with respect to unamplified nodes.

1.1.6.1 Setup

To demonstrate that the advantages of the RF front-end in a real multi-hop application help to contain the overall power consumption of the network, we ran a testbed in which we compared the MagoNode CC2950 and a common unamplified mote, the IRIS. To do so, we deployed two sensor networks in



	TX	RX	IDLE
🟪 EK1	14.6	12.9	13.2
🟩 TelosB	19.1	22.4	20.8
🟥 IRIS	21.4	18	19
🟦 Magonode CC2590	27.7	14.5	16.9

Figure 1.4: Current consumption comparison between the MagoNode and unamplified motes. EK1 is comparable to a MagoNode without the RF front-end

the basement of our department: the first one made of MagoNode CC2950 and the second one composed by IRIS motes (figure 6.5). Like many indoor scenarios the location is characterized by obstacles that shrink the radio range of the nodes. In the specific, the basement of our building features 70 cm walls thickness and metal doors.

Each testbed is made of 20 nodes powered by 2x 1.5 V AA alkaline batteries and ran for one day. We used the Collection Tree Protocol (CTP) [56,57] as routing layer and Low Power Listening (LPL) [55] as MAC, both implemented in TinyOS 2.1.2 distribution. In particular, we exploited the RFXLink library which enforces a common radio stack for both platforms. We tuned CTP and LPL using the parameters listed in table 1.3 (please refer to [56] and [55] for parameters details). Each testbed ran for 24 hours, each node transmitted a data packet toward the sink (Node 0) with an Inter-Packet Interval (IPI) of 5 minutes, using channel 11 of IEEE 802.15.4. We activated the TrafficMonitor-Layer of the RFXlink library which enables detailed network statistics related to the radio activity such as the overall radio active time, the time spent in

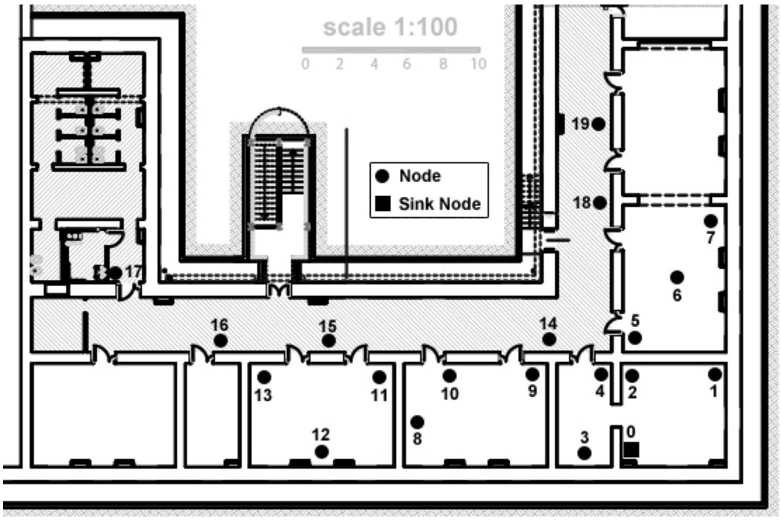


Figure 1.5: Testbed topology.

transmission and reception, the overall number of bytes transmitted and received. All these values are carried by each transmitted data packet which has a length of 103 bytes.

Table 1.3: Testbed parameters.

Parameters	Value	Parameters	Value
Inter-Packet Interval	300 s	CTP SENDDONE OK OFFSET	500 ms
Battery	2x 1.5 V	CTP SENDDONE OK WINDOW	250 ms
Packet Size	103bytes	CTP LOOPY OFFSET	4 s
Run Time	24 h	CTP LOOPY WINDOW	4 s
Nodes Number	20	CTP NO ROUTE RETRY	20 s
LPL Wakeup Period	500 ms	CTP MIN INTERVAL	1 s
LPL Listen	10 ms	CTP MAX INTERVAL	500 s
LPL Delay After Receive	20 ms	RF Channel	11

1.1.6.2 Metrics

We compared both testbeds in terms of network reliability, generated traffic and power consumption. We evaluated the network reliability statistics through the following metrics:

- **Data Delivery Ratio (DDR):** this is the ratio between data packets transmitted by each nodes and actually received by the sink.

- **Total Parent Chgs**: CTP automatically selects the parent of a node based on the link reliability. This metric represents the overall number of parent changes of the whole network.

On the other side, the generated traffic statistics were calculated exploiting the following metrics:

- **Total ReTx:** CTP features automatic re-transmission of unacknowledged data packets. The overall count is aggregated in this metric.
- **Average Time Has Lived (THL):** each data packet carries the hop count represented by the THL metric provided by CTP. The average THL is then an average routing-level network diameter, i.e. the average number of hops a packet has to traverse before reaching the sink node.
- **Total Fwd:** this metric represents the overall number of data packets that have been forwarded over the whole multi-hop network.

Finally, we computed the power consumption statistics, derived from TrafficMonitorLayer, exploiting the following metrics:

- **Total CC [TX, RX, IDLE]**: these metrics point out the overall current consumption of the network in mJ when nodes are respectively in TX, RX and IDLE state.
- **Total TS [TX, RX, IDLE]**: these metrics point out the overall time spent by nodes in TX, RX and IDLE state.

1.1.6.3 Results

Table 1.4 shows that the reliability is excellent for both network, however the MagoNode generates much less traffic in terms of data re-transmission, forwarded packets and average THL to achieve this results. This is due to the higher transmission power provided by the RF front-end, that extends the radio range of the nodes and strengthens the reliability of the channels. Remarkably, this entails a minor use of the radio that leads to a lower overall power consumption. Indeed, as we can observe from table 1.4, the sum of TX, RX and IDLE time of the MagoNode network is sensibly lower with regard to IRIS network, as well as the current consumption. In particular, while consumption of the MagoNode network in both TX and RX are close to the IRIS ones, in IDLE, the MagoNode network reduces the consumption by about 20%. Note that the higher TX time of the IRIS platform also affects the IDLE time. This is caused by the LPL transmission mechanism, which consists in a burst of packets interleaved by backoff and ack-wait periods. During those periods the transceiver is in idle listening state. We also notice how the IDLE power consumption of both networks is more than 90% of the total one, hence, it represents by far the predominant consumption when comparing networks performance. This observation validates the design choice discussed in section 1.1.3.1 that keeps the RX/IDLE overhead as low as possible.

Table 1.4: Testbed results. Δs are calculated with respect to IRIS values.

	MagoNode	**IRIS**	Δ (%)
Avg DDR (%)	100	100	0
Tot Parent Chgs	38	114	-66
Tot ReTx	319	1393	-77
Avg THL	1.55	2.14	-28
Tot Fwd	3395	6804	-50
Tot CC TX (mJ)	130379	132655	-1.7
Tot CC RX (mJ)	37973	39391	-3.5
Tot CC IDLE (mJ)	1713799	2139719	-20
Tot TS TX(s)	1569	2066	-24
Tot TS RX (s)	873	729	+2
Tot TS IDLE (s)	33803	37538	-10

1.1.7 Concluding Remarks

The MagoNode platform outperforms other RF front-end platforms in terms of energy consumption. In particular, we explained how a flexible platform, that could be equipped with the proper front-end depending on the specific geographical context, helps to keep the current consumption low in many application scenarios. We also compared the MagoNode performance to common unamplified motes showing that current consumption in TX, RX and IDLE are comparable.

The obvious advantages introduced by an RF front-end such as extended radio range and more reliable links, allows the MagoNode to better fit to many real scenarios, like structural health monitoring, where traditional unamplified motes requires several additional relay nodes to guarantee the appropriate network connectivity. To support this argument, we compared the MagoNode to the IRIS, a widely spread unamplified mote, in a real multi-hop indoor scenario. Results, reported in table 1.3, show that the average THL (Time Has Lived), i.e. the number of hops a packet must traverse in order to reach the sink, is 1.3 times higher for the network made of IRIS. Moreover in the IRIS testbed, the overall energy consumption is driven by a high amount of forwarded and retransmitted packets. Vice versa in the MagoNode case, the benefits introduced by the RF front-end shrink the number of overhead packets lowering the overall network traffic and the consequent waste of energy.

1.2 Wake-Up Radio

Emerging WSN applications demand longer system lifetimes and stringent latency constraints, contrasting goals that are difficult to obtain together. Indeed, in order to minimize transmission delay, there is the need to make use of the radio apparatus intensively. Since the radio transceiver is the most power demanding hardware module in a WSN mote, it is mandatory to minimize the energy consumption of radio communication, in order to maximize the lifetime of the network.

Several techniques have been proposed to mitigate the power consumption due to radio communications [58]. As observed in the previous section, these techniques aim to reduce to idle listening that is known to be the main source of energy waste in WSN [59]. The idle listening is defined as the time a mote has to listen to the channel waiting for incoming messages. A common technique to reduce idle listening is to duty cycle the radio activities which means to periodically switch on and off the transceiver. This approach allows to increase the lifetime of the network to months or a year. However, while helping in power saving, duty cycling increases the latency of radio communications since a mote is not always ready to receive due to the alternation of radio on/off periods. Moreover, duty cycling does not eliminate completely the idle listening since there will always be an amount of time in which the transceiver had to be turned on and listen to the channel for a short amount of time.

To overcame these difficulties, research has recently shifted towards the design of communication architectures for WSNs based on wake-up radios. A wake-up receiver (WRx) is an hardware module which continuously listen to the channel and wakes the mote main transceiver only when needed. In this way the main transceiver can stay off until a message has to be actually received or transmitted. Since a WRx continuously listen to the channel a real power saving can be achieved only if its power consumption is orders of magnitude lower that the one of the main transceiver. It is thus mandatory for WRx to be ultra-low power, consuming only hundreds of nA while listening to the channel. This can be achieved only using passive, or almost passive, state-of-the-art electronic components. Moreover, a WRx has to feature high sensitivity, robustness to interference and low latency. The sensitivity is highly correlated to the communication range: the higher the sensitivity (measured as the capability to sense the weakest signal in -dBm), the lower the range. Improving the sensitivity provides a straight-forward way to increase the performance of the WRx. However, it should to be noticed that increasing the sensitivity often implies an increment of power consumption, so there is always the need to find a trade-off between sensitivity and power consumption.

The outstanding energy saving introduced by wake-up radios comes at the price of shorter radio range compared to traditional active radio transceivers.

Indeed, a state-of-the-art wake-up system can achieve a range of tens of meters, while common active radio apparatus can reach up to hundreds of meters. This seeming contradiction can be addressed in two way: adapting actual communication protocols for WSN, which need to be re-engineered to exploit the advantages of wake-up technology limiting its drawbacks, or designing new ones which natively supports radio wake-up (in paragraph "Future Work" of section 6.6 we propose a new communication protocol thought to take advantage of WUR benefits).

In the remainder of this section we discuss the state-of-the-art of wake-up receivers (subsection 1.2.1). Then, we detail the architecture of our ultra low-power WRx (subsection 1.2.2). Finally, in subsection 1.2.3, we introduce the MNA Wake-Up Board, an extension board designed for the MagoNode featuring a complete WUR system made of our WRx prototype and a 868 MHz wake-up radio transmitter (WTx).

1.2.1 Related Works

A variety of methods and techniques have been recently proposed to design completely passive or low power wake-up radio technologies. The first wake-up radio design was presented by Gu and Stankovic in [60], and consists of a passive radio-triggered circuit able to harvest power from the received radio signal and to use the stored energy to generate a wake-up interrupt. The proposed approach reaches a wake-up range of around 3 meters, which makes it suitable for short-range applications such as Wireless Body Area Networks (WBANs), but hinders its practical use in typical WSN applications that require long range communications.

Some efforts have explored the use of RF-powered RFID tags to realize passive wake-up receivers with [61] and without [62] addressing capability. This class of solutions have wake-up ranges that are typical of UHF RFID systems, i.e., around 10 m [63]. However, an RFID reader is needed to transmit wake-up sequences with output power in the order of Watts. For example, a 12 m range is possible by transmitting at 4 W EIRP power [64]. While this approach is promising for some application scenarios, it is generally unsuitable for WSN applications based on peer-to-peer networking, since in passive RFID-based radio-triggering circuits the energy toll shifts to the transmitter side.

Due to the limitations of passive designs, research efforts have recently focused on semi-active circuits requiring only a minimal fraction of the receiver components to be supplied with external power. The goal is that of realizing wake-up receivers with high sensitivity (and thus extended wake-up range) and ultra-low power consumption. Several solutions use an envelope detector implemented through Schottky diodes, MOSFETs or ad hoc ICs for the radio front-end, and a comparator to generate an interrupt. Works following this approach present wake-up receivers with a sensitivity between -35 and -47 dBm and a power consumption in the range from 2.3 and 10 μW [65–67]. Despite being a significant step further towards low-power, high range wake-up radios, these solutions have a lower sensitivity and require a significantly higher power consumption than our proposed design.

Recent research has proposed receivers with nano-power consumption. These solutions, however, generally suffer from limited wake-up ranges, therefore failing to meet the needs of general sensing systems applications. For example, a remarkably low power consumption of 98 nW is achieved by the wake-up receiver presented by Roberts and Wentzloff, which operates at 915 MHz using OOK modulation [68]. However, this design is shown to wake up node at most 1.2 m when using a transmission power of 0 dBm and patch antennas for communication. Another solution featuring an extremely low power 270 nW radio-triggering circuit designed for WBAN applications was presented by Marinkovic and Popovici [69]. Their receiver uses Gaussian On-Off Keying (GOOK) pulse width modulation and reaches an operating range of no more than 10 m at +10 dBm transmission power output with a 0 dBi antenna.

1.2.2 Wake-Up Radio Receiver

In this section we detail the design of a wake-up radio receiver (WRx) with high sensitivity, fast reactivity, very low power consumption, and selective addressing capability. The receiver uses on-off keyed (OOK) modulation, the simplest form of amplitude-shift keying (ASK) modulation in which digital data are represented as the presence or absence of a carrier wave. The proposed architecture is made up of four main blocks: a matching network, the envelope detector, the interrupt generator, and an optional addressing module (figure 1.6).

The matching network provides maximum power transfer between the antenna and the rest of the circuit by supplying a stable impedance match to the input source. It consists of an LC network (see figure 1.6), whose inductor and capacitor values need to be adjusted accordingly to the transmission frequency. In this way, the WRx can be tuned to work at any specific frequency depending on the applications needs. The output of the matching network is an RF signal.

The second block is a passive demodulation circuit that allows to recover the information content from the modulated signal. As our design uses OOK modulation, the demodulation circuit consists of a passive envelope detector that discards the frequency and phase content of the RF signal and only detects its amplitude. In our prototype implementation, we used a single-stage half-wave rectifier with series diodes. We selected the HSMS-285C diodes from Avago Technologies, which are optimized for sub-GHz frequencies and for incoming power lower than -20 dBm. Once the signal is rectified, the bits of the received wake-up sequence are reconstructed by using an ultra-low power comparator. To perform this task, we used an adaptive threshold mechanism that keeps the V^{-} pin of the comparator at half of the input signal level. With this approach, we use the energy from the antenna, rather than a voltage divider, for generating the threshold, thus reducing the static power consumption of the circuit. The adaptive mechanism is designed based on a simple RC circuit (R1-C4 in figure 1.6) that is connected to the negative input of the comparator. The positive input of the comparator is directly connected to the signal from the voltage multiplier, which allows the detection of both weak and strong signals. The voltage offset of the comparator directly affects the overall sensitivity of the wake-up radio: with a lower voltage offset it is possible to sense smaller signals. However, comparators with lower voltage offset generally have higher current consumption. In our prototype, we selected the TLV3701ID comparator from Texas Instruments [70], which features a reasonable energy consumption of 579 nA. Exploiting above cited hardware components, our WRx can achieve a receiving sensitivity up to of -50 dBm.

The successive stage of the circuit is the preamble detector. To filter interference from noise or other communications that can generate unwanted awakenings, we use a specific preamble in the wake-up packet that can be detected by this passive part of the circuit. The preamble is an OOK modulated signal featuring a specific length (i.e. number of bits) that is sent at a specific bit rate f_p. On-off keyed modulated signals that are sent at a data rate lower than f_p are filtered out by the RC filter (R3-C5 in figure 1.6), which works as a low-pass filter. If a preamble with the specific length and frequency is received, the preamble detector signals a wake-up interrupt to the on-board microcontroller, which then starts reading data from the pin connected to the output of the comparator (before the diode SD3).The microcontroller performs address matching, waking up the main node only when a valid wake-up address is received. The on-board microcontroller is optional, indeed address matching operations can be performed directly by the mote main microcontroller.

“

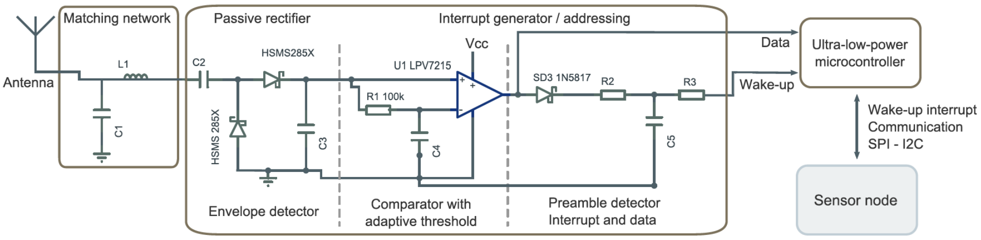


Figure 1.6: WUR architecture

1.2.3 MNA Wake-Up Board

We integrated our wake-up architecture tuned at 868 MHz with the MagoNode (introduced in section 1.1). The wake-up extension board designed for the MagoNode, named MNA Wake-Up Board, includes two modules: the WRx prototype presented in the previous paragraph and a 868 MHz wake-up radio transmitter (WTx). Both modules feature also a 868 MHz antenna with a 50Ω impedance and 2 dBi gain. We choose to use separate antennas for the WRx and the WTx for greater accuracy in assessing the performance of our receiver. Although desirable from an integration perspective, sharing a single antenna would require an additional antenna switch that introduces non-negligible losses, whence a negative impact on the link power budget. The impedance matching of the RF section was carefully designed to provide the maximum power transfer at 868 MHz between the antenna and the rest of the circuit. Details on the tuning of the matching network have been presented in [71]. Wake-up interrupts generated by the WRx are sent directly to the MagoNode by connecting the output of the comparator to a low-level asynchronous interrupt pin of the mote microcontroller. In such case the optional on-board microcontroller is not installed, meaning that signal decoding and address matching duties are carried out by the MagoNode. To decode a received signal the MagoNode executes an oversampling decoding routing, which samples each received bit of a wake-up message multiple times with a certain frequency. Then, it decides if the bit is a logic one or a logic zero basing on values assumed by the majority of samples.

Wake-up sequences are transmitted by using the WTx module, which is implemented exploiting the Anaren A1101R08C OEM module. The A1101R08C features the CC1101 transceiver from Texas Instruments, a low-power sub-1GHz transceiver which supports several modulation formats, including OOK. Communication with the MagoNode is realized through the SPI interface. In our application the CC1101 is configured to exploit the 868 MHz band, with an output power of +10 dBm. Figure 1.7 shows a snapshot of our Wake-Up Radio enabled WSN mote.

1.3 Wake-Up Radio Experimental Characterization

In this section we present the result of the experimental activity we carried on in order to characterize the performance of the WRx prototype in terms of power consumption, sensitivity and radio range.

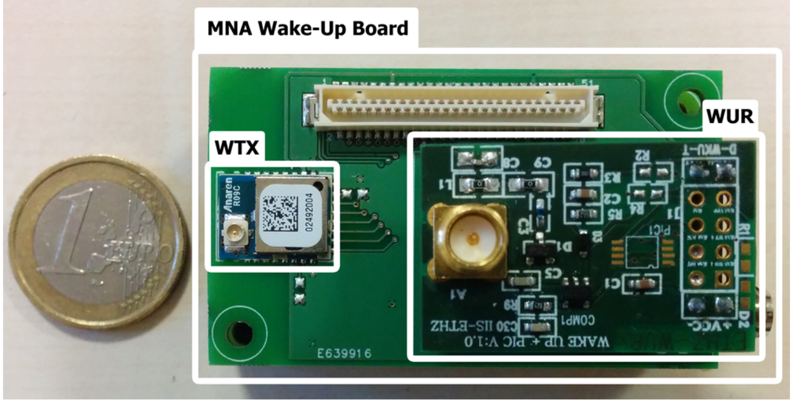


Figure 1.7: The MNA Wake-Up Board

1.3.1 Power Consumption

To determine the minimum power consumption of the WRx, we supplied the prototype with 1.8 V and we measured the current using a shunt resistor of 100 Ω. The result is a power consumption of 1.276 μW. To better characterize the power consumption profile of our WRx prototype, we also measured the current consumption of the WRx in idle listening and while receiving high bits (i.e. logic one). To do so, we connected a digital multimeter in series to a WRx prototype supplied with 3 V from a DC power supply. Then, we programmed a Magonode equipped with a MNA Wake-Up Board to send two different wake-up sequences. Figure 1.8 shows results for sequences 10110100 and 11111111. The measured current consumption is around 579 nA in idle listening and 48 μA when receiving a logic one (recall that the power consumption of the MagoNode in reception is 14.5 mA). It is important to notice that all consumption measurement were performed using a WRx prototype without the optional on-board microcontroller, which involves an overhead in terms of energy consumption.

1.3.2 Sensitivity

To characterize the sensitivity of the WRx in a controlled environment, we set up a testbed consisting of an attenuator connected to two Magonode equipped with a MNA Wake-Up Board (a transmitter and a receiver). The attenuator is used to reduce the power of the signal emitted by the transmitter, so as to emulate the attenuation experienced by a wake-up packet transmitted over different distances. We tested the maximum sensitivity of the WRx by continuously sending wake-up sequences consisting of one byte set to 11111111 (hexadecimal 0xFF), while increasing the attenuation of the transmitted sig-

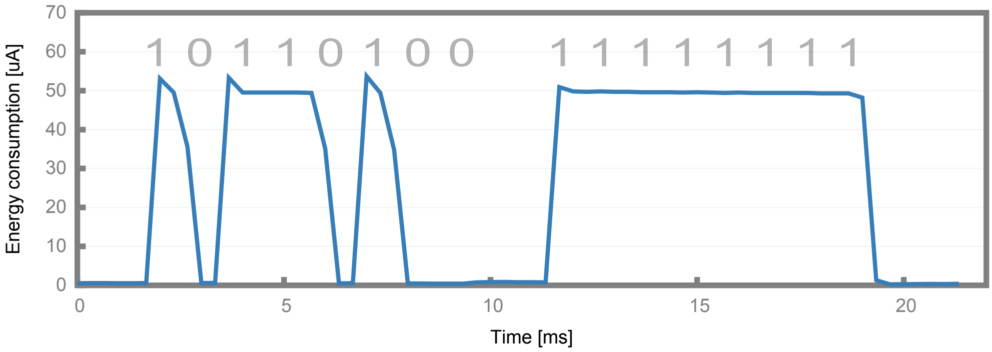


Figure 1.8: Current consumption of the WRx when receiving two different sequences.

nal. The measured sensitivity, which is the biggest value of attenuation at which the wake-up receiver was still able to generate an interrupt, was determined to be −50 dBm.

1.3.3 Radio Range

In this set of experiments we perform an empirical evaluation of the maximum wake-up range varying the distance between two MagoNode featuring the MNA Wake-Up Board. One node acted as *Transmitter* and was programmed to send a message consisting of one byte set to 0xAA every second at 1 kbps, while the other acted as *Receiver*. Both nodes were equipped with a 2 dBi antenna. The goal of these experiments was to assess the maximum distance at which the WRx is able to receive the whole wake-up message without errors. Tests were performed both in an indoor scenario consisting of a room $15m \times 20m \times 6m$ $(W \times D \times H)$ and in an outdoor open-field, placing *Transmitter* and *Receiver* in Line Of Sight (LOS).

By progressively increasing the distance between the two nodes, we measure a maximum outdoor range of 15 m and a maximum indoor range of 19 m. The difference between indoor and outdoor range can be explained from the fact that in indoor environments waves are guided by walls while nodes are place in LOS. To empirically prove that we performed the very same test in a corridor 3 meter wide and 2,5 meters high: the maximum range achieved was 27 meters. Then, we explored the performance of the WRx in an outdoor scenario while receiving a series of 88 1-byte wake-up sequences. To do so, we deployed a *Transmitter* *Receiver* in the courtyard of the building of the Department of Computer, Control, and Management Engineering of University of Rome “La Sapienza”. The *Transmitter* was programmed to send 100 messages for each sequence with a inter packet interval (IPI) of 100 ms. The

Receiver was programmed to continuously listen to the channel and, each time it overhears traffic, to store the received message with its relative timestamp into a persistent memory buffer.

It is important to notice that in real contexts the performance of the WRx is affected not only by the distance between *Transmitter* and *Receiver*, but also from environmental and RF noise. Keeping that in mind, received messages were labeled as follows:

- ***OK***: if all bits of the message are correct and if its relative timestamp is equal to IPI.
- ***Wrong***: if the message has one or more bits wrong and if its relative timestamp is equal to IPI.
- ***Noise***: if the message has one or more bits wrong and if its relative timestamp is different from IPI.
- ***Lost***: if message reception is jeopardized by noise.

We progressively increase the distance between the two nodes and we evaluated the fraction of wake-up sequences correctly received and recognized at different distances. We repeated the test five times for distance, then we calculated the average of all runs. Results of these experiments are shown by Figure 1.9. For distances up to 12 meters, selective awakenings are very reliable: The percentage of WRx packets correctly received is always higher than 99.64%. When the distance between the transmitter and the receiver is set to 12 meters, 98.81% of WRx packets are correctly received. At this distance, the percentage of lost packets and of packets received incorrectly due to data corruption is less than 0.54% and 0.65%, respectively. For longer distances, however, performance of selective wake-ups worsen. At 15 meters, less than half of the packet is correctly received, and the percentage of lost packets reaches 6.7%.

1.4 Wake-Up Radio Noise Characterization

In radio communications the reception of signals is impaired by noise perceived by the radio receiver. Noise affecting radio systems is the overlap of white noise, radio frequency interference (RFI), thermal noise, environmental noise and multipath fading. According International Telecommunication Union (ITU) Radio Regulations [72], radio frequency interference is defined as “The effect of unwanted energy due to one or a combination of emissions, radiations, or inductions upon reception in a radio communication system, manifested by any performance degradation, misinterpretation, or loss of information which could be extracted in the absence of such unwanted energy”. Most

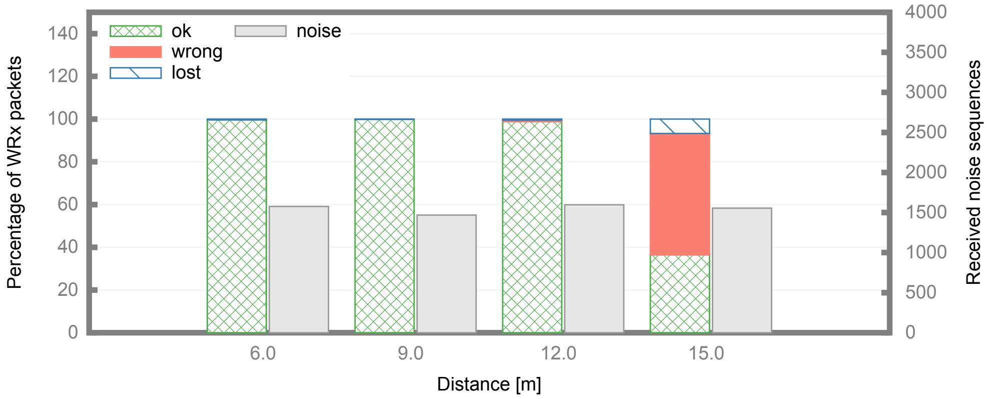


Figure 1.9: Percentage of received WRx packets at increasing distances and noise sequences received.

electrical or electronic equipment can produce RFI. Typical items which can cause interference are electric motors, car ignition systems, switching power supplies, relays, and equipment such as electronic printers, medical instrumentation, industrial controls, personal computers and electronic games. These types of problems can be tracked down and eliminated. Moreover, RFI can be caused by other radio apparatus. Indeed, radio transmitters emit RF signals at frequencies other than their characteristic frequency. These undesired signals, known as out-of-band emissions, may overlap with the typical frequency of a radio system causing performance degradation and an increase in data error rate. Some types of natural processes, such as lightning discharge and multipath fading, can affect electromagnetic waves propagation. Multipath fading happens when a radio signal reaches the receiver by means of two or more paths, causing destructive interference. It can be caused by reflection from terrestrial objects, human artifacts and water bodies and by atmospheric phenomena, such as skip and ducting. Skip is caused by the ionization of gases in the ionosphere, which reflect radio waves for long distances causing harmful interference. Ducting occurs when air of different temperatures and humidity forms layers in the lower atmosphere. These layers reflect electromagnetic waves which propagate much farther than usual. Finally, RFI can be caused by thermal noise caused by thermal agitation of electrons within an electrical circuit. It occurs regardless of the applied voltage since electrons vibrate as a result of the temperature.

Noise perceived by the WRx affects its performance in terms of communication reliability, radio range and power consumption causing spurious awakenings of the receiver and errors in received wake-up messages. A spurious awakening happens when the waking of a mote occurs as result of a noise event perceived by the WRx. This phenomenon adversely affects motes power consumption since they wake-up in time instants in which they were supposed to stay in sleep mode. On the other side, noise overlapping to wake-up packets can cause bit flipping and thus errors in received messages. Direct consequence of this behavior is a drop in packet delivery ratio and in radio channel reliability.

Because of above cited issues and to dimension the wake-up radio performance depending on the application scenario, it is mandatory to quantify the noise level perceived by the WRx and to understand features characterizing noisy events (e.g. length in time of noise spike and maximum and minimum length of noise burst).

1.4.1 Noise Level Quantification

A common metric used to quantify the noise level in WSN application contexts is the received signal strength indicator (RSSI). The noise level is measured by means of a mote continuously sampling the RSSI, when the wireless channel is clear. The noise level estimation accuracy depends on the RSSI sampling frequency: the higher the sampling frequency, the higher the accuracy. In [73] and [74] experiments about noise level estimation involving high frequency RSSI measurements were presented. In particular, in [73] TelosB motes were used to measure the RSSI with a frequency of 1 kHz and the retrieved measurements were stored in the flash memory of the mote. Instead, in [74] a EMSPCC11 mote [75] equipped with the CC1100 IEEE 802.15.4 transceiver was used to sample the RSSI value with a frequency of 5 kHz and measured samples were sent to a PC through the UART interface. High-frequency RSSI sampling could be difficult, especially when the measurement is done using devices with limited computational capabilities and low memory capacities like WSN motes. For example, using the CC1101 transceiver featured by our WTx module1.2, the minimum operation time required to read a RSSI and store it into a RAM buffer is 125 μs, which translates into a maximum RSSI sampling frequency of 8 kHz. This implies that noise spike shorter than 125 μs have a chance to be undetected. Moreover, taking into account the amount the amount of RAM memory of the MagoNode (16 KB) and subtracting the amount of memory required by the application, the RAM buffer can contain at most 15000 one byte entries. So at the maximum RSSI sampling frequency after 1.875 s the RAM buffer is completely full. This is the maximum time span in which it is possible to continuously sample the RSSI at 8 kHz. To prologue time span we tried to write measured samples data into the flash memory of the MagoNode, which features a capacity of 2 MB. Anyway, this is not feasible in practice because write to flash operation execution time is much higher than write to RAM. Furthermore, we need to consider that for a continuous sampling we need to keep the CPU of the microcontroller and the transceiver in active state, which implies a power consumption in the order of tens of mA.

Because of such drawbacks and limitations we decided to use the Bit Error Rate (BER) metric instead of the RSSI to quantify the noise level perceived by our WRx. To understand why BER is a suitable metric for the quantification of the noise perceived by the WRx, let us consider the ideal case: in an environment without noise and without any transmission, a WRx continuously listening to the channel receives only low bits (logic zeros). Assuming the baud rate to be 1 kbaudps (1 ms for each bit), in a given time of t milliseconds a WRx will receive t zeros. On the other hand, in a non-ideal environment, in a given time of t milliseconds the WRx will receives some high bits due to noise. Counting these bits and dividing them by t will give the BER induced by noise perceived by WRx. To actually measure the BER induced by noise in a given environment we used a MagoNode equipped with a WRx continuously listen to the channel. Every time the WRx is triggered by a noise event, the MagoNode decodes the received signal and logs received bits into a persistent memory. Then, data collected on the field are analyzed offline and an estimate of the BER induced by noise is computed. The measurement can be repeated various times in order to increase the noise level evaluation accuracy.

The above described noise evaluation procedure has computational complexity and memory requirements which are trivial compared to those needed by the RSSI measurement routine. Indeed, the MagoNode stays in sleep state while no noise events are detected. Moreover, using the BER metric, there is no need to correlate RSSI measurements to noise events perceived by the WRx, easing the offline analysis of data collected on the field.

As an example, in table 1.5 we report the results of a noise quantification experiment in an outdoor environment. The experiment consisted of three runs in which a MagoNode equipped with our WRx collected data about noise level for 21 s, 154 s and 1048 s respectively. Results show an average BER induced by noise is $2,71 * 10^{-1}$, which is high, considering that 10^{-2} is considered acceptable for packetized voice application over the a wireless medium and 10^{-5} is regarded as acceptable for uncoded data [76]. We performed the noise level evaluation procedure in five different environments: two office rooms

Table 1.5: Noise level measured in a outdoor high noise environment.

	Running Time(s)	Total Noise Bits	BER
Run1	21	6936	$3,3 * 10^{-1}$
Run2	154	42980	$2,79 * 10^{-1}$
Run3	1048	212014	$2,02 * 10^{-1}$

with a Wi-Fi access point each, a lecture hall with four long range Wi-Fi access point, the courtyard of our department (DIAG) in Rome and the roof of the ETZ building of ETH Zürich, which is characterized by various antennas and radio apparatus transmitting in the sub 1GHz and the 2.4 GHz band. For each environment we repeated a noise evaluation procedure lasting 15 minutes for five times, then we average the obtained results which are reported in table 1.6.

Table 1.6: Average noise level measured in various environments.

Environment	Avg Noise Level (BER)
Office 1	$6,2 * 10^{-4}$
Office 2	$5,1 * 10^{-4}$
DIAG courtyard	$1,19 * 10^{-3}$
Lecture Hall	$1,1 * 10^{-2}$
ETZ roof	$3,4 * 10^{-1}$

From these experimental results we inferred a rough classification of environments based on noise level perceived by the WRx:

- *noise level in the order of* 10^{-1}: high noise environment, radio wake-up communications are almost jammed by noise

- *noise level in the order of* 10^{-2}: noisy environment, radio wake-up communications are severely affected by noise

- *noise level in the order of* 10^{-3}: low noise environment, radio wake-up communications are partially affected by noise

- *noise level in the order of* 10^{-4}: noise free environment, radio wake-up communications are affected by noise in a way that is not relevant in practice

1.4.2 Noise Signals Features

Knowing the noise level characterizing an application scenario is not enough to properly tune wake-up radio performance. There is also the need to understand features of noise signals causing spurious awakening and errors in received wake-up messages. In the specific, we want to classify noise events depending on their length in time and so observe if interference appears in form of spikes or bursts. This is mandatory to properly tune up the radio wake-up system since different form of noise affect WRx performance in different ways. Indeed, a fast noise spike can cause a spurious awakening of the

WRx but it is not likely to cause a bit flipping, while a burst can affect several bits of a wake-up message causing data corruption. Since our wake-up system was designed to work at 1 kbps, we assumed that a spike is a noise event which lasts an amount of time less or equal to a bit time (1 ms at 1 kbps) while a burst is an event which lasts at more than 1 ms. Among spikes events we distinguished between fast spikes (shorter than 0.5 ms) and spikes lasting between 0.5 ms and 1 ms, while within bursts we classified noise events depending on their length in milliseconds. In this ways a *1-to-2* burst is noise event lasting between 1 ms and 2 ms, a *2-to-3* burst is a noise event lasting between 2 ms and 3 ms and so on...

We analyzed shape of noise signals collected during the above cited noise quantification experiments. As it possible to see in table 1.7 we computed the percentage of each noise event type: without considering the ETZ roof scenario, where the noise level is so high to prevent wake-up communications, in the other environments the maximum burst length is 3 ms (i.e. 3 bit times at 1 kbps). Moreover, the percentage of fast spikes is almost twice the one of spikes lasting between 0.5 ms and 1 ms in both office environments, while it is higher than about 10% in the DIAG courtyard and the lecture hall.

Table 1.7: Percentage of noise event types measured in various environments.

Environment	Fast Spikes	Spikes	*1-to-2* Bursts	*2-to-3* Bursts	Longer Bursts
Office 1	63.94%	29.74%	6.32%	0%	0%
Office 2	59.41%	31.37%	9.22%	0%	0%
DIAG courtyard	47.63%	35.68%	15.31%	1.38%	0%
Lecture Hall	30.27%	40.91%	26.56%	2.26%	0%
ETZ roof	10.74%	18.69%	37.32%	21.13%	12.12%

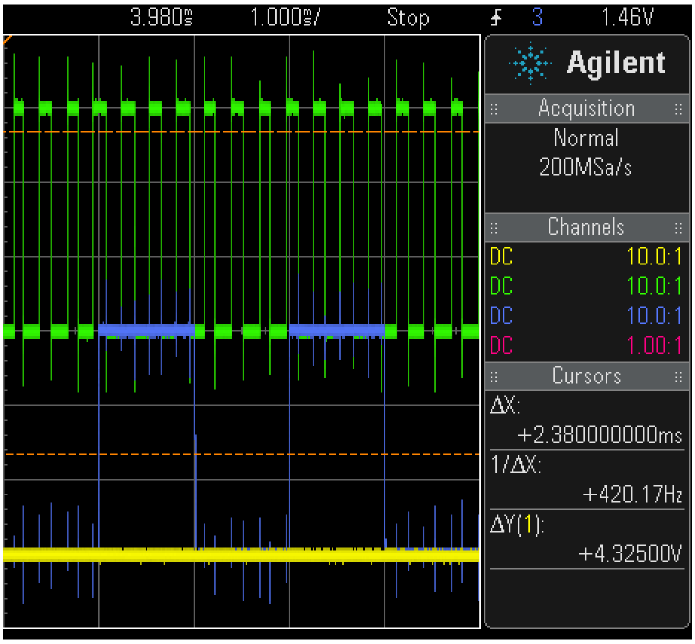
1.5 Wake-Up Radio Noise Avoidance Mechanisms

Taking into account results came out from experimental tests for characterization of noise signals (see section 1.4), we equipped our WRx with noise avoidance mechanisms in order to mitigate the impact of noise interference on radio performance. In particular, we implemented a hardware preamble mechanism to filter out noise bursts and a software oversampling routine to filter out fast noise spikes.

1.5.1 Oversampling Decoding Routine

The oversampling decoding method samples each bit of a wake-up message multiple times with a certain frequency. Then, it decides if the bit is a logic one or a logic zero basing on values assumed by the majority of samples.

Figure 1.10 shows the oversampling routine decoding 4 bits (0101) received by out WRx. Each toggle of the green probe corresponds to an instant at which a sample value is taken, while the blue probe represents the signal received by the WRx. Accordingly to results presented in table 1.7 fast noise spikes (i.e. noise events lasting less than 0.5 ms) represent a relevant portion of noise perceived by our WRx in various indoor and outdoor environments. To filter that kind of noise we decided program the oversampling routine to take 7 samples per bit. Remembering that our wake-up radio system is tune at 1 kbps (i.e. 1 ms per bit), one sample lasts about 0.142 ms and four samples form the majority set. So, in order to decided if a bit is a logic one, received signal has to stay high for an amount of time greater or equals to 0.568 ms (i.e. the amount of time required to take 4 samples). In this way all fast noise spike are completely filtered without affecting radio wake-up performance. This is one of the main advantage of the oversampling routine. On the other side, this decoding method is a energy consuming task since the microcontroller has to stay in the active state for the whole duration of the procedure.



3.980ms 1.000ms/ Stop 3 1.46V

Agilent

Acquisition
Normal
200MSa/s

Channels
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 1.00:1

Cursors
ΔX:
+2.380000000ms
1/ΔX:
+420.17Hz
ΔY(1):
+4.32500V

Figure 1.10: Oversampling Routine decoding 4 bits (0101)

1.5.2 Hardware Preamble Detector

To avoid spurious awakenings caused by noise, we implemented a hardware preamble detector mechanism which is able to trigger a wake-up interrupt only when a predefined sequence is detected by the WRx. The hardware preamble is implemented by a RC filter connected at the output of the comparator, on a side, and to a interrupt pin of the microcontroller, on the other (see figure 1.6). The circuit logic scheme is reported in figure 1.11. When a high bit (i.e. a logic one) is received by the WRx the output of the comparator goes high, charging the capacitor C through the resistance R. After an amount of time t the voltage level on C rises above the high level input voltage (VIH) of the microcontroller triggering an interrupt. t is given by the following equation, where VCC is the system supply voltage:

$$t = -[(R * C) * (\log \frac{V_{CC} - V_{IH}}{V_{CC}})]$$

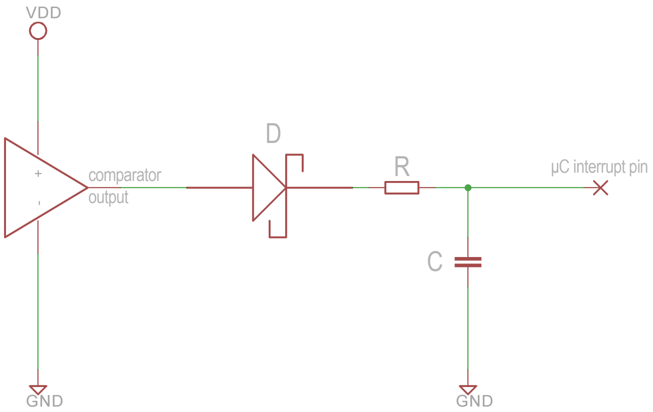
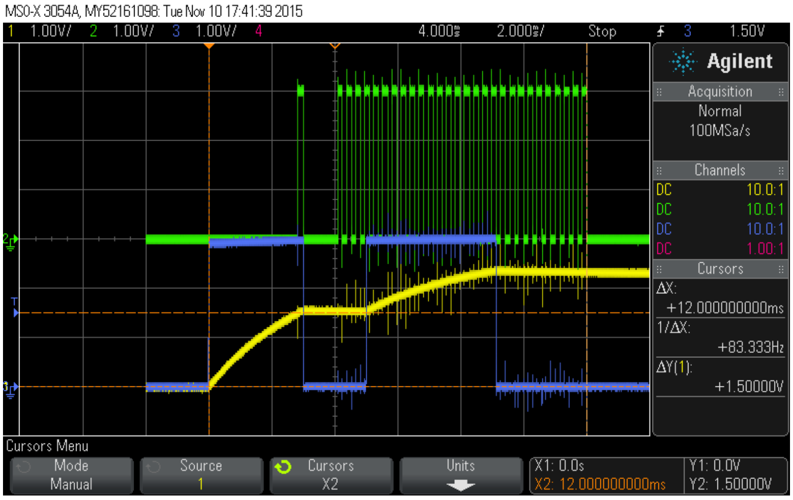


Figure 1.11: Preamble detector circuit logic scheme.

Taking into account the maximum noise burst experimentally computed in paragraph 1.4.2, we decided to set the hardware preamble in such a way to trigger the interrupt only if the comparator output stays high for at least 2.6 ms. Then we set the transmitter to send a preamble equals to 111 which has length 3 ms at 1 kbps. We decided to oversize the preamble length in order to take into account hardware components tolerances, which could led to a slight variation of the value of t calculated by above cited equation.

The first design of the hardware preamble detector did not take into account the time-to-discharge of the capacitor C. This is an important parameter since it is the lower bound on the inter packet interval (IPI) limiting wake-up packets transmission rate. Indeed, if two packets were sent with a IPI lower than the capacitor time-to-discharge, the reception of the second packet would begin when the capacitor has still a certain amount of charge. This would result in a shorter time to trigger the microcontroller interrupt. When it happens the noise filtering capabilities of the hardware preamble are reduced. Now let us suppose to deploy the radio wake-up system in an environment with a high noise level. In this case noise signals received by the WRx will increase the amount of charge of C, shortening the time to trigger the interrupt. In the worst case the noise does not allow the capacitor to discharge. When this happens the reception of a new wake-up message fails since the trigger does not fire because the voltage across C is already above VIH.

Figure 1.12 shows a complete reception of a message with payload equals to 0x78 with a preamble detector implemented without the *RDown*. The yellow probe was connected to the microcontroller interrupt pin (trigger pin); the blue probe was connected to the comparator output before the preamble detector (data pin); the green probe was connected to a debug pin which toggles when the interrupt on the trigger pin fires at the end of the preamble and every time a sample is taken by the oversampling routine. The grid was set to 2 ms per division. As we can see, when the WRx begin to receive the preamble (blue probe) the capacitor C starts the charging phase (yellow probe). When the voltage across C reaches VIH, the interrupt fires and the debug pin (green probe) toggles. When the preamble ends the debug pin toggles again. After that, the system waits for 1 ms, then it begins the oversampling routine to decode the message. From picture 1.12 we can notice that C is charged also by high bits of the payload and that the voltage across its terminals is above VIH after the end of the message reception. The C discharge time problem


MSO-X 3054A, MY52161098: Tue Nov 10 17:41:39 2015

1 1.00V/ 2 1.00V/ 3 1.00V/ 4 4.000ms 2.000ms/ Stop 3 1.50V

Acquisition: Normal 100MSa/s

Channels: DC 10.0:1, DC 10.0:1, DC 10.0:1, DC 1.00:1

Cursors: ΔX: +12.000000000ms; 1/ΔX: +83.333Hz; ΔY(1): +1.50000V

Cursors Menu: Mode Manual; Source 1; Cursors X2; Units

X1: 0.0s X2: 12.000000000ms Y1: 0.0V Y2: 1.50000V

Figure 1.12: Reception of a wake-up message without *RDown*

was completely solved adding a pull-down resistor (*RDown*) in series to the microcontroller interrupt pin and fixing the behavior of the wake-up firmware. The corrected logic scheme of the hardware preamble is shown in figure 1.13. During the preamble reception the interrupt pin is put in high impedance

(Hi-Z) state. This allow$C$ to be charged through R avoiding the current to flow through $RDown$. As soon as the voltage across C goes above VIH the interrupt fires. In the interrupt service routine (ISR) the interrupt pin is grounded, allowing the capacitor to discharge through $RDown$. The interrupt pin is kept in this state until the end of the message reception, avoiding C to be charged by received bits. After that, when the system is able to receive a new message, the interrupt pin is set in the Hi-Z state again. If $RDown$ is wisely chosen the discharge is fast enough IPI is limited only data rate of the wake-up system. Picture 1.14 shows a complete reception of a message with a preamble detector implemented with $RDown$ resistor. Probes are connected as in picture 1.12, while the grid was set to 1 ms per division. As we can see, as soon as the trigger pin interrupt fires, the capacitor C is discharged by the resistor $RDown$. Moreover, it is possible to notice that high bits of the message do not charge$C$, since during the reception of the payload the current flows through $RDown$ to ground.

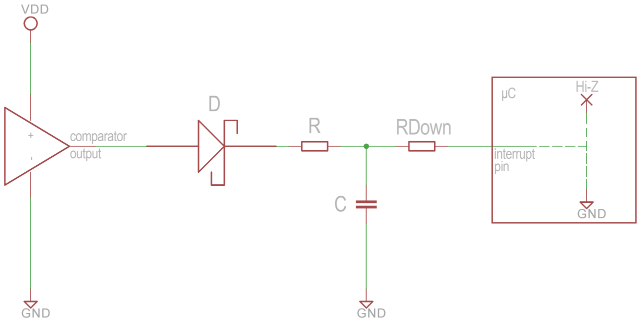
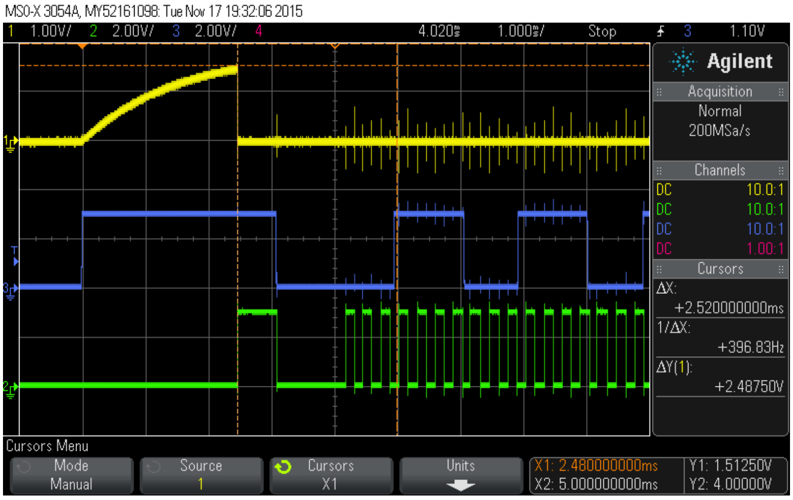


Figure 1.13: Preamble detector circuit logic scheme with the pull-down resistor.

1.6 Wake-Up Radio Error Correction

As already pointed out in section 1.4, noise perceived by our WRx adversely affects wake-up radio communication reliability, achievable radio range and power consumption. In the specific, noise signals overlapping to wake-up packets can cause bit flipping and thus errors in received messages. Direct consequence of this behavior is a drop in packet delivery ratio. Another limitation affecting radio wake-up performance is given by the inability of the WRx to correctly receive wake-up messages featuring bit sequences with more than six consecutive “1”. This behavior is direct consequence of the adaptive threshold mechanism implemented in our WRx architecture. Indeed, if this mechanism has the fair advantage of to be very power efficient (see section



MSO-X 3054A, MY52161098: Tue Nov 17 19:32:06 2015

1 1.00V/ 2 2.00V/ 3 2.00V/ 4 4.020ms 1.000ms/ Stop 3 1.10V

Acquisition
Normal
200MSa/s

Channels
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 1.00:1

Cursors
ΔX: +2.520000000ms
1/ΔX: +396.83Hz
ΔY(1): +2.48750V

Cursors Menu
Mode: Manual
Source: 1
Cursors: X1
Units

X1: 2.480000000ms
X2: 5.000000000ms
Y1: 1.51250V
Y2: 4.00000V

Figure 1.14: Reception of a wake-up message with *RDown*

1.2.2), on the other side, it has this important limitation.

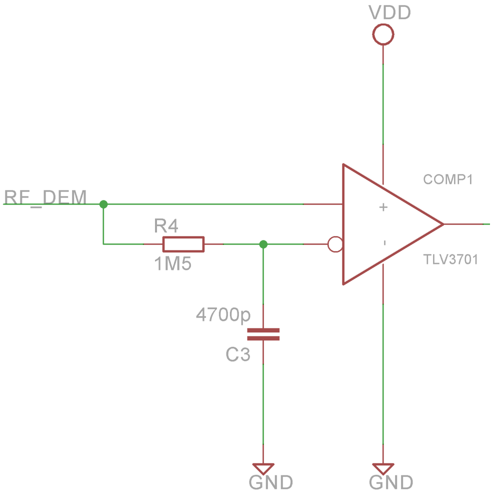


Figure 1.15: Adaptive threshold mechanism circuit logic scheme.

To explain why our WRx is not able to receive more that six consecutive logic "1", let recall that the bits of the received wake-up packets are reconstructed using an ultra-low power comparator. To perform this task, the adaptive threshold mechanism that keeps the V- pin of the comparator at half of the input signal level is used (see section 1.2.2). The adaptive mechanism is implemented by a simple RC circuit (R4 and C3 in figure 1.15) that is connected to the negative input of the comparator, while the positive input is directly connected to the signal from the voltage multiplier (see figure 1.6). When the WRx receives a high bit (i.e. a logic “1”), the voltage on V+ became equals to the input signal level, while voltage on V- pin became equals to the voltage across *C3* which is charged from the current flowing through *R4*. Then, the comparator compares voltage values on V+ and V-: since the WRx is receiving a “1”, V+ voltage will be higher than the one on V- and the comparator will outputs a “1”. Now, let suppose that the WRx receives an infinite sequence of high bit. After a certain number of received “1”, which depends on *R4* and *C3* values, the amount of charge across *C4*, and so the voltage on pin V-, will become higher than input voltage level on V+ pin. In this case the comparator will output a logic zero even if the transmitted bit is a logic one.

1.6.1 Error Correction Codes

Forward Error Correction (FEC) is a technique to ensure robustness and reliability of data transmissions adding redundant information to the original information by means of an Error Correction Code (ECC). This redundancy affects maximum achievable data rate of the system and has an overhead in terms of energy consumption of communications. Indeed, it is trivial to say that using more bits to send the same amount of information implies a reduction of the overall system data rate. Likewise, transmitting redundancy bits has a obvious cost in terms of energy consumed by communication apparatus, especially in systems characterized by low baud rates. In fact, these systems require a higher time to send a bit, compared to system characterized by higher transmission speed.

FEC codes are divided in block codes and convolutional codes [77]. A block code (n,k) encodes messages made of k data symbols in blocks (code words) of n symbols. If symbols are represented by bits, the block code is binary. Among binary block codes there are linear block codes. A block code is linear if each code word can be formed by the modulo-2 sum of two or more code words. Linear block codes feature error detection and correction capabilities based on minimum distance between code words (Hamming distance): a linear block code featuring a distance d can detect up to $d-1$ and correct up to $d-1/2$ errors [78]. Linear block codes are easy to implement and feature a low coding delay.

Convolutional codes feature powerful error correcting capabilities, but are more complicated and more difficult to implement than block codes. They are extensively used in many applications such as digital video, satellite communications, mobile communications, etc. Convolutional codes are hard to decode since the encoding procedure is based in a finite state machine which features branching paths to encode bits of a given data sequence. A very efficient decoding procedure for convolutional codes based on the maximum-likelihood method is the Viterbi Algorithm.

For radio wake-up purpose we took into account only linear block codes so far. In the specific, we concentrate our attention on Hamming codes. Hamming codes are single error correcting codes (SEC), meaning that only one error can be corrected per block of information. We decided to focus on SEC codes like Hamming's, instead of codes capable of correcting bursts of errors, since results reported in table 1.7 show that in four of the five considered application scenario, the majority of noise events recorded are spikes or fast spikes. In particular, in both offices noise spikes are more than 90% of all noise events, while in the courtyard and in the lecture hall this percentage exceeds respectively 70% and 80% of overall noise perceived by our WRx. Only in the roof scenario bursts events overcome noise spikes by 40%. Anyway, as already pointed out in section 1.4.1, roof scenario is a borderline case characterized by a high noise level which severely impairs wake-up radio communications.

1.6.1.1 Hamming Codes

Being SEC codes, all Hamming codes feature a minimum distance between code words equals to 3.Error correction capabilities of Hamming codes are based on the addition of parity bits to the original information. A parity bit is a redundant bit that forces a binary string to have a specific parity, which comes in two varieties even and odd. A string of bits has even parity if it contains an even number of bits equal to 1, otherwise it has odd parity. The parity bit for an even parity block may be computed by summing all the bits modulo-2. The parity bit for an odd parity block may be computed by summing all the bits modulo-2 and adding 1.

A linear block code (*n,k*), where n is code words length and k is the number of data bits, belongs to the Hamming codes family if for every m integer greater or equal to two $n = 2^m - 1$ and $k = 2^m - m - 1$. Where m is the number of parity bits. Hamming codes are perfect codes, indeed all bits sequences not belonging to the code have distance 1 from any code word. Defining the code rate as the ratio between data bits and parity, hamming codes achieve the highest possible rate given the number of parity bits (see table 1.8).

A strategy to build a Hamming code *H(n,k)* is to design the encoder in such a way parity bits are placed in power-of-two positions of the code word, while data bits occupy the remaining positions. In this way bit in position 1, 2, 4, 8, etc., are parity bits. Putting parity bits positions in binary form, we can notice that they feature only one bit equals to 1: 1 is 1b, 2 is 10b, 4 is 100b, 8 is 1000b etc... On the other side all positions which binary form features more than one bit equals to 1 are data positions. If the result of the

Total Bits	Parity Bits	Data Bits	Code Rate
3	2	1	0,333
7	3	4	0,571
15	4	11	0,733
31	5	26	0,839
$2^m - 1$	m	$2^m - m - 1$	$\frac{2^m-m-1}{2^m-1}$

Table 1.8: Hamming codes rate.

logic AND between the binary form of a code word position and the binary form of a parity bit position is equals to 1, then the considered parity bit is responsible for that code word position. In such way, the parity bit in position 1 (1b) is responsible for code word positions 1 (1b), 3 (11b), 5 (101b), 7 (111b) etc... The parity bit in position 2 (10b) is responsible for the following bit positions: 2 (10b), 3 (11b), 6 (110b), 7 (111b) and so on. To check for an error there is the need to compute for each parity bit the XOR between values in bit positions belonging to that parity bit (expect the parity bit itself). The result of this operation outputs a binary value which gives the position of the bit in error.

Hamming codes can be represented in matrix form using a matrix **G** of dimension $k \times n$ called generator matrix. Columns of the generator matrix are made of column vectors representing data bits and columns vectors representing parity bits. The column vector $\hat{d_i}$ representing the i-th data bit is made of all zeros and a one in i-th position. The column vector $\hat{p_i}$ representing the i-th parity bit contains a one for each data bits the considered parity bit is responsible for and a zero in all other positions. Columns of the generator matrix can be arranged in several ways depending on needs. For example a common practice is to put $\hat{d}$ vectors in the first k positions of the **G** matrix and $\hat{p}$ vectors in the remaining $n - k$ positions. A Hamming code represented by such **G** is called systematic. To encode a data sequence of k bits it is enough to compute the product by the line vector representing the data sequence and the matrix **G**.

An efficient way to reveal an error in a received bits sequence is to use parity check matrix **H**. The parity check matrix has dimension $k \times (n-k)$ and feature the following property: $\mathbf{G} \cdot \mathbf{H}^T = 0$. Multiplying the line vector $\hat{y}$ representing the received code word by $\mathbf{H}^T$ we obtain a vector called syndrome. If the syndrome associated to the vector $\hat{y}$ is zero then$\hat{y}$ is a code word, otherwise an error occurred. It is important to notice that the syndrome vector allow only to reveal the presence of an error in a received sequence, but not to locate it. To find the position of the bit in error there is the need to locate the column of the matrix **H** matching the syndrome associated to the received sequence.

Flipping the bit in that position will result in a valid code word.

As already pointed out, Hamming codes can correct a single error and detect a double error, but they cannot discriminate between a single error in a code word and a double error of another. In few words, when a Hamming code reveals the presence of errors in a code word, it means that there at most two errors in that code word. An extra parity bit can be added to Hamming codes in order to prevent this drawback. The extra parity bit (or overall parity bit) check the parity on all the bits in the code word. Adding the overall parity bit the distance between code word increases to 4, which means that the extended Hamming code can correct single errors, can distinguish between single and double errors and can reveal up to three errors. Extended Hamming codes belong to the SEC-DED (Single Error Correction, Double Error Detection) family.

Encoding and decoding operations of a Hamming codes can be implemented in two ways: computing matrices multiplications or using look-up tables containing encoded and decoded values computed offline. The former approach requires a higher computational capabilities than the latter, which has constant computational complexity. Indeed, encoding operation using matrices require a multiplication between the data vector and the matrix $\mathbf{G}$ associated to the Hamming code, while the decoding procedure requires a matrix multiplication between the received sequence vector and the matrix $\mathbf{H}$. On the other side, the matrices multiplication approach has little memory requirements, while storing look-up tables need a memory space which is proportional to the number of data and parity bits making up code words. In the specific, there is the need to store two look-up table. The first one maps code words in decoded values and contains 2^n entries of $\lceil \frac{n}{8} \rceil$ bytes each. The second table maps data sequences in code words and contains 2^k entries of $\lceil \frac{k}{8} \rceil$ bytes each.

From WSN point of view, it is possible to implement a Hamming code on a low-power microcontroller in a very efficient way exploiting look-up tables using a limited number of bits to represent a code word. The upper bound of code word length depends on the amount of RAM (Random Access Memory) the microcontroller feature. For example,*H(7,4)* is suitable for microcontroller featuring limited amount of memory, since its look-up tables occupy 144 bytes: 2^7 bytes for the decoding table and 2^4 bytes for the encoding table. Employing a limited number of bits to represent code words has a drawback in terms of data rate the code can achieve. Indeed, having a look to table 1.8 it is possible to notice that increasing code words length is proportional to the code rate. On the other side, increasing the code rate implies a reduction of the error correcting capabilities given a fixed amount of transmitted bits. To see that, let suppose a system has to transmit 44 data bits over a noisy channel. Using

H(7,4) data bits are encoded in 11 code words and the transmitter has to send a total of 77 bits. In this case the receiver can eventually correct up to 11 errors. On the contrary, using *H(15,11)* data bits are encoded in 4 code words and the transmitter has to send a total of 60 bits. In this case the receiver can eventually correct up to 4 errors.

1.6.2 Limiting Consecutive Ones in Wake-Up Sequences

To limit the number of consecutive "1" in a wake-up sequences several encoding can be applied to the original message. Among the others: the Pulse Interval Encoding (PIE), the Manchester encoding and the 8b/10b encoding. To assure that encoded sequences feature a limited number of consecutive "1", all these encoding systems add some redundancy to original data, thus reducing the data rate of the system and adding an overhead in terms of energy consumption.

In PIE high bits of the original message are encoded with the sequence 100 while low bits are encoded with the sequence 10. The main advantage of the PIE encoding is that it that a high bit is always followed by a low bit, limiting to one the number of consecutive high bits. On the other side, the number of bits of an encoded sequence depends on counts of ones and zeros in the original sequence. Let suppose to encode with PIE 8 bits sequences: the lower bound of encoded sequence length is 16, while the upper bound is 24. This behavior also influence the maximum data rate achievable by a system using PIE: in the worst case the overall bit rate is reduced by three times.

In the Manchester encoding, also known as Phase Encoding (PE), each bit is transmitted in a fixed period and a is represented by a low-to-high or a high to low transition. Transitions occur in the period midpoint. PE limits the number of consecutive high bits in an encoded sequence to two and allow encoded sequences to feature the same number of ones and zeros (i.e. encoded sequences are DC-balanced). This advantages comes at expenses of the maximum data rate achievable by the system which is reduced by two times, compared with the case without encoding.

In the 8b/10b coding scheme 8 bits data word are split in two subsequences of 5 bits and 3 bits, respectively. Then, 5 bits subsequences are encoded in 6 bits blocks, while 3 bits subsequences are encoded in 4 bits blocks. Concatenating encoded groups results in a 10 bits code word which is ready to be transmitted over the channel. Since 10 bits are used to encode 8 bits data words, some of the possible 2^{10} combinations are not used. Properly choosing code words among available combinations it is possible to limit the number of consecutive bits in an encoded sequence. In the standard implementation, 8b/10b coding scheme ensures that encoded sequences features at most five consecutive high bits (i.e. ones). Moreover, exploiting unused combinations, it is possible to represent data words with more than one code word in order to statistically ensure DC-balanced encoded sequences. In the standard implementation, 8b/10b coding scheme ensures DC-balanced code words encoding some of the data subsequences in two different blocks of opposite disparity. In general the disparity of a bit string is computed counting the number of ones and subtracting the number of zeros. If a bit string has disparity equals to zero it is called neutral, otherwise it features positive disparity if the number of ones is strictly grater than the number of zero or negative disparity vice versa. So, if all 6 bits and 4 bits encoded blocks are disparity neutral, the 10 bits code words are disparity neutral too and the resulting code is perfectly DC-balanced. Anyway, since among 16 possible combinations of 4 bits encoded blocks only 6 are disparity neutral, it is not possible to encode all the 8 values of the 3 bits data subsequences with DC-balanced sequences. Likewise, since among 32 possible combinations of 5 bits encoded blocks only 20 are disparity neutral, it is not possible enough to encode all the 32 values of the 5 bits data subsequences with DC-balanced sequences. So there is the need to use block with positive or negative disparity in order to represent all data subsequences. Because both the 4-bit and 6-bit blocks have an even number of bits, the disparity positive disparity will be always +2 and the negative disparity will be always -2. Of course, using unbalanced encoded blocks, the resulting bit stream transmitted over the channel will be DC-unbalanced. To overcome this issue the 8b/10b encoding scheme was designed to allow data subsequences which are not associated to disparity neutral blocks to be encoded with two blocks featuring opposite disparity (i.e. one block with disparity +2, the opposite with disparity -2) and to combine 4 bits encoded blocks with 6 bits encoded blocks in such a way that the worst case disparity featured by a 10 bits code word is at most +2 or -2. In such a way, the encoder can choose the appropriate code word depending on the current Running Disparity (RD), which is zero, negative or positive accordingly to the disparity of the previously transmitted code word, so limiting the number of unbalanced bits in a stream to at most two. Using the 8b/10b coding scheme the data rate is reduced of 20% compared with the case without encoding. The 8b/10b encoding can be easily implemented using a lookup tables which map data sequences in encoded sequences. The lookup table implementation has the fair advantage to feature a negligible computational complexity. On the other side, the cost in terms of memory space required to store look-up table is not negligible. Anyway the size of lookup table can be managed by state-of-the-art micro-controller which features tens of KB of RAM memory (see subsection 1.6.4 for details).

1.6.3 MagoNode Wake-Up Radio Encoding System

The first version of the MagoNode wake-up radio firmware featured 8 bits wake-up addresses, meaning that up to 256 different nodes can be uniquely identified, and a double encoding system to transmit information in a robust and reliable way. Given a wake-up address, the encoding procedure worked as follows: first of all, the wake-up address was split in two nibbles (i.e. two data sequences of 4 bits) and encoded using a SEC-DED Hamming code *H(8, 4)*, which is able to correct one error, to distinguish between single and double errors and to reveal up to three errors in a received code word. Then, Hamming code words were encoded using 8b/10b encoding, which limits to at most five the number of consecutive ones in an encoded sequence. Overall, the double encoding system was able to encode a 8 bits wake-up address in a 20 bits code sequences.

Both Hamming and 8b/10b codes were implemented using look-up tables. Hamming code *H(8, 4)* required two tables: one 16 bytes table for encoding and one 128 bytes table for decoding. The 8b/10b encoding procedure required two pairs of tables: one to encode 3 bits data subsequences, the other to encode 5 bits data subsequences. A table of each pair contained positive disparity encoded blocks, while the other held negative disparity blocks (see subsection 1.6.2 for details). The total amount of occupied memory was 80 bytes. On the other side, the 8b/10b decoding procedure required two tables: one 16 bytes table to decode 4 bits blocks and one 64 bytes table to decode 6 bits blocks, for a total of 80 bytes memory occupation. The overall memory space occupied by encoding and decoding tables of both coding schemes was 304 bytes. This value is completely manageable by the MagoNode which features a microcontroller (Atmega256RFR2) with 32 KB of RAM memory.

The drawback of the double encoding system is that it heavily shrinks the data rate of the system which is reduced to 60% of the rate without encoding. To mitigate this drawback, the actual version of the MagoNode wake-up radio firmware features an encoding system which is based only on a Hamming code *H(8, 4)*. Indeed, arranging properly columns of the generator matrix **G**, it is possible to create a Hamming code *H(8, 4)* featuring code words with at most four consecutive bits, thus eliminating the need of the 8b/10b encoding. In this way, we created an encoding system able to correct and detect errors and to limit the number of consecutive one in a encoded wake-up address using just an Hamming code *H(8, 4)*. Doing so, the actual version MagoNode wake-up radio firmware is able to encode 8 bits addresses in 16 bits code sequences, saving both energy and time required to transmit a wake-up message. Furthermore, the data rate reduction caused by redundant bits is 50% of the rate without encoding, 10% less that the first version of the MagoNode firmware. Finally, the amount of memory occupied by the new version of the firmware is just

144 bytes, more than a half less than the previous version. Anyway to achieve this result there is a price to pay: the number of code words which can be used to encode data nibbles is reduced from 16 to 15. This drawback has a direct impact on the number of 8 bits wake-up sequences that can be used to address node, which is changed from 256 to 226. Anyway, this is a reasonable price to pay, keeping in mind that number of nodes characterizing application scenarios envisioned for wake-up enabled WSN platforms is in between ten to hundred. nodes. In table 1.9 are reported data nibbles and corresponding Hamming code words featuring the Hamming code *H(8, 4)* implemented in the actual version of the MagoNode wake-up radio firmware.

Table 1.9: Data Nibbles and corresponding Hamming Code Words

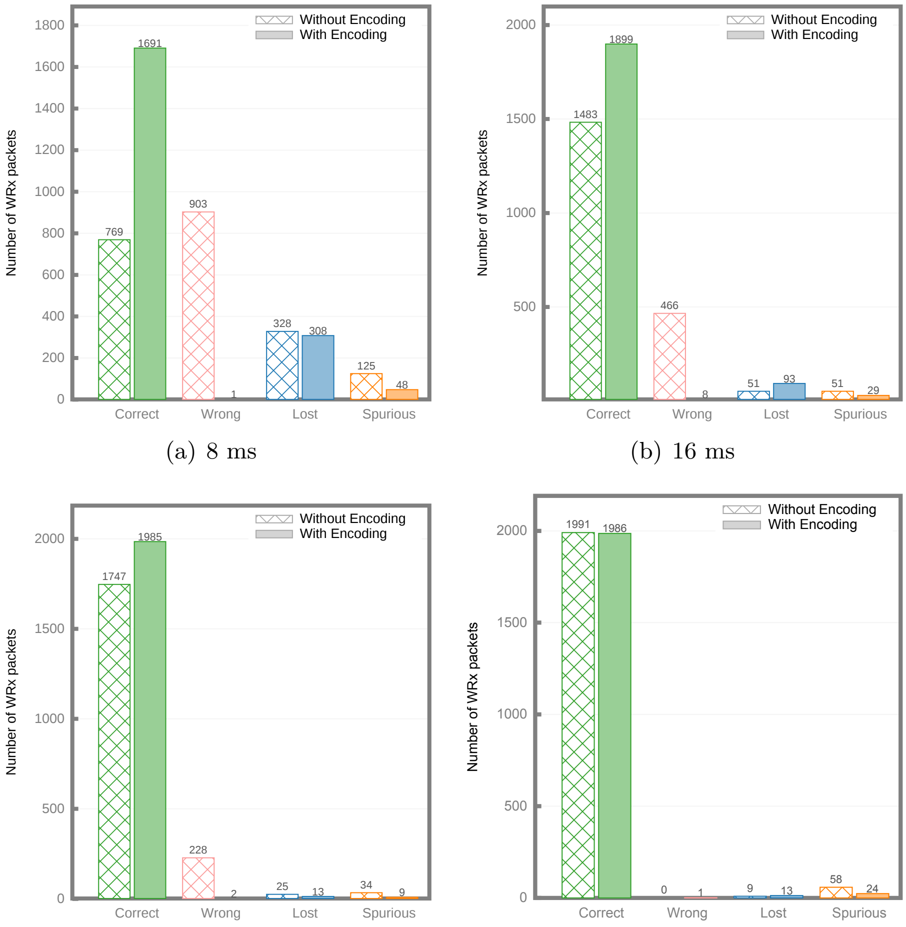
Data Nibbles	Code Words	Code Words	Pared Code Words
0	0x00	0000000	00000000
1	0x71	1110001	01110001
2	0x62	1100010	11100010
3	0x13	0010011	10010011
4	0x54	1010100	11010100
5	0x25	0100101	10100101
6	0x36	0110110	00110110
7	0x47	1000111	01000111
8	0x38	0111000	10111000
9	0x49	1001001	11001001
A	0x5A	1011010	01011010
B	0x2B	0101011	00101011
C	0x6C	1101100	01101100
D	0x1D	0011101	00011101
E	0x0E	0001110	10001110

1.6.4 Performance of the Encoding

As we already pointed out in subsections , the advantages introduced by an encoding system in terms of robustness and reliability of radio communication come at expenses of maximum achievable data rate and bring an overhead in terms of energy consumption of radio apparatus. These drawbacks are a fair price to pay in order to keep data delivery ratio high and packets retransmissions low in environments featuring high BER induced by noise signals. Anyway, when the BER induced by noise drops and deployment environment tends to be noise-free, it is reasonable to assume that there is no need for an encoding system and that its cost it is an unmotivated overhead in terms of energy consumption and data rate reduction. To prove this hypothesis and to find a boundary in terms of BER induced by noise above which it is worth to use the encoding system and below which it is better to exploit code-free communications, we deployed an experimental testbed and we conducted a series of experiments. The testbed was made of three MagoNode equipped with the MNA Wake-Up Board (see section 1.2.3): one acting as transmitter, another acting as receiver, the last one acting as noise source. For each test, the transmitter sent 10 different wake-up sequences of length equal to one byte. Wake-up sequences are sent 100 times each with a Inter Packet Interval (IPI) of 100 ms. This procedure was repeated two times for a total of 2000 messages sent per test. The transmission power was set to +10 dBm. The receiver stored all received data in a persistent memory and was placed 3 meter away from the transmitter. The noise source was placed 15 cm away from the receiver and was programmed to periodically send an interference sequence at -10 dBm of power. Each test we performed featured a different noise pattern and was repeated with and without the encoding system. Since conducted experiments featured an artificial noise source, we deployed the testbed in an almost noise-free indoor environment which was characterized by a BER induced by noise of 10^{-5}. We did that to avoid environmental noise to influence the outcomes of tests.

Figure 1.16 shows the results of such tests, in which the MagoNode acting as noise source was programmed to send one noise bit every 8, 16, 32, 128 ms. When a noise bit is generated every 8 ms, the BER induced by the artificial noise source was 10^{-1}. Outcome of the test, reported in Figure 1.16(a), shows that exploiting MagoNode encoding system featuring the hamming code *H(8, 4)* (see paragraph 1.6.4) bring to a packet delivery ratio of 84.55%, while sending messages without encoding (raw bits) brings to a packet delivery ratio of only 38.45%. Therefore, the Packet Error Rate (PER) using the encoding system is almost four time less with respect to the one measured without exploiting the encoding. Thus, when the BER induced by noise is in the order of 10^{-1} it is definitively worth to use the MagoNode encoding system. When a noise bit is generated every 16 ms, the BER induced by the artificial noise source was $6 * 10^{-2}$. Outcome of the test, reported in figure 1.16(b), shows that exploiting MagoNode encoding system featuring the hamming code *H(8, 4)* bring to a packet delivery ratio of 94,95%, while sending messages without encoding (raw bits) brings to a packet delivery ratio of only 75,15%. Therefore, the Packet Error Rate (PER) using the encoding system is almost 5 time less with respect to the one measured without exploiting the encoding. When a noise bit is generated every 32 ms, the BER induced by the artificial noise source was $3 * 10^{-2}$. Outcome of the test, in figure 1.16(c), shows that exploiting MagoNode encoding system featuring the hamming code *H(8, 4)* bring to a packet delivery ratio of 99,25%, while sending messages without encoding (raw bits) brings to a packet delivery ratio of only 87,35%. Therefore, the Packet Error Rate (PER) using the encoding system is almost 17 time less with respect to the one measured without exploiting the encoding. We can conclude that when the BER induced by noise is in the order of $3 * 10^{-2}$ it is still worth to use the MagoNode encoding system. However, the performance of the system with and without encoding is near to 90% of packets correctly delivered. When noise bits are generated more sparsely, using encoding is definitively not worth since it does not bring to a performance gain in terms of PER compared to the case without encoding. In particular, if a noise bit is generated every 128 bits, results, reported in figure 1.16(d), show the packet delivery ratio is consistently higher than 99.3% regardless of the encoding being used.

As a second set of tests, we program the node acting as noise source to send a noise pattern made of a burst of bits every 32, 256 and 512 ms. The experiments were repeated three times: the first time with burst length equals to two (noise pattern 0xC), the second time with burst length equals to three (noise pattern 0xE), the third time with burst length equals to four (noise pattern 0xF). It is important to notice that the WRx embedded in the MNA Wake-Up Board features a preamble detector of three bits (see paragraph 1.5.2), so noise burst featuring three or more high bits will cause a spurious awakening with high probability. Figure 1.17 reports the results of our experiments when the generated noise pattern is set to 0xC, i.e., a sequence containing two consecutive bits of noise. When interference sequences are sent every 32 ms, exploiting the MagoNode encoding system brings to a packet delivery ratio of 91.1%, while sending messages without encoding (raw bits) brings to a packet delivery ratio of only 78.7% (figure 1.17(a)). When noise bursts are sent less frequently (i.e., every 256 ms or 512 ms, as reported by figure 1.17(b) and 1.17(c), respectively) the packet delivery ratio is higher than 97% both with and without the encoding system, so there is no need to use the encoding system in such cases. Thus, using the encoding system give a substantial performance advantage when noise bursts are rather frequent, but it does not provide enhanced performance when the frequency of noise bursts is low. Figure 1.18 reports the results of our experiments when the generated noise pattern is set to 0xE, i.e., a sequence containing three consecutive bits of noise. When a noise burst is sent every 32 ms, the packet delivery ratio is way below 50% both with and without the encoding (figure 1.18(a)). Moreover, the number of spurious awakenings is very high: 5511 without encoding, 3921 using the encoding system. This outcome can be explained considering the fact that a noise burst of 3 bits has a high probability to cause a spurious awakening and remarks the necessity to set the hardware preamble according to the maximum burst length measured in the deployment environment in order to avoid performance drop in terms of PER (see paragraph 1.5.2 for further details). Finally, it is possible to notice that the data delivery ratio sending raw bits messages is 14% higher than using the encoding system. This is due to the additional overhead introduced by the encoding, which doubles the time to receive an encoded sequence. Thus, the probability that a noise burst overlaps to the reception of a message, causing bit flipping and errors, is higher when using the encoding system. When the burst frequency is relatively low (one burst every 256 ms or 512 ms), noise has a limited impact on performance both with and without message encoding. In such cases, the packet delivery ratio is always above 90% (figure 1.18(b) and figure 1.18(c)). Figure 1.19 reports the results of our experiments when the generated noise pattern is set to 0xF, i.e., a sequence containing four consecutive bits of noise. Considerations made for experiments with noise pattern 0xE hold also in this such. The only difference is given by a proportional decrease of packet delivery ratio at every sending frequency, both with and without encoding. This behavior can be explained by a higher number of noise bits characterizing the noise pattern 0xF compared to the number of noise bits characterizing 0xE.



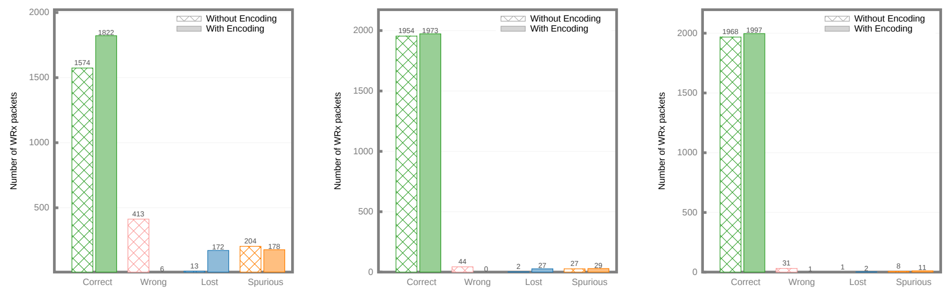
(a) 8 ms

(b) 16 ms

(c) 32 ms

(d) 128 ms

Figure 1.16: Number of correct, wrong, lost and spurious WRx packets received without and with encoding for single bits of interference generated periodically every 8, 16, 32 and 128 ms.

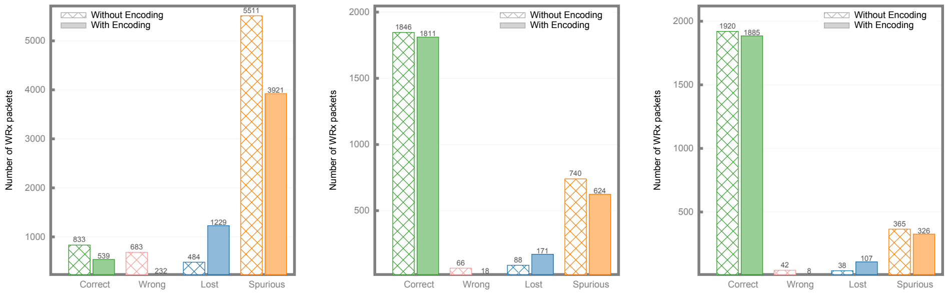


(a) 32 ms

(b) 256 ms

(c) 512 ms

Figure 1.17: Number of correct, wrong, lost and spurious WRx packets received without and with encoding for two bits of interference (packet 0xC0) generated periodically every 32, 256 and 512 ms.

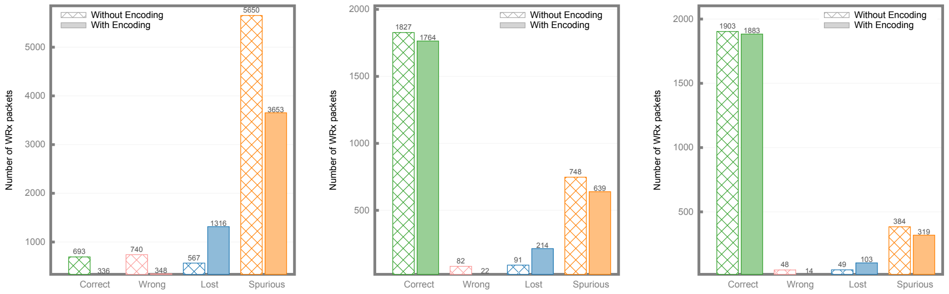


(a) 32 ms

(b) 256 ms

(c) 512 ms

Figure 1.18: Number of correct, wrong, lost and spurious WRx packets received without and with encoding for three bits of interference (packet 0xE0) generated periodically every 32, 256 and 512 ms.



(a) 32 ms

(b) 256 ms

(c) 512 ms

Figure 1.19: Number of correct, wrong, lost and spurious WRx packets received without and with encoding for three bits of interference (packet 0xF0) generated periodically every 32, 256 and 512 ms.

Chapter 2

Energy-Aware Protocols

Energy efficiency of low-power hardware components is not enough to guarantee motes lifetime required by most WSN applications. To overcome this issue there is the need to control low-power hardware featured by sensor nodes with energy-aware software layers able to optimize the energy usage to accomplish typical WSN tasks (transmit, receive, sense, etc.). In this scope, an energy-aware wireless sensor nodes is an embedded device where the energy efficiency of the hardware platform is furtherly increased by software mechanisms with the ability to dynamically calculate a trade-off between energy consumption, system performance and operational reliability. Furthermore, it is important to point out that energy-aware solutions not only concern the optimization of the power consumption of a single mote, but also the maximization of the lifetime of the whole network. In such case energy-aware software solutions for WSNs become distributed algorithms sharing information about energy consumption on the whole network, instead of running on a single mote.

In chapter 2 we describe our efforts to improve the energy efficiency of protocols dedicated to WSNs introducing energy aware solutions. In particular, in section 2.1 we introduce REACTIVE, an energy-aware protocol, which is able to boost the energy efficiency of Deluge T2, an over-the-air programming (OAP) protocol implemented in TinyOS 2.x [27]. Furthermore, in section 2.2 we present the ALBA-WUR protocol, a cross-layer solution for data gathering in WSNs that redesigns ALBA-R protocol [29], increasing its energy efficiency exploiting our wake-up radio system (see section 1.2).

2.1 Over-the-Air Programming

Over-the-air programming (OAP) for Wireless Sensor Networks (WSNs) is an essential feature for long-term applications, where physical access to deployed wireless motes cannot be guaranteed or when time and costs of manual reprogramming exceed the benefits. Typically, OAP protocols for WSNs are based on broadcast dissemination of a new image or part of it [79–82]. Because of its inherent broadcast nature and of the large amount of data disseminated over the network, OAP is an energy consuming task. Commonly used metrics for evaluating the energy efficiency of an OAP protocol are the disseminated image size and the dissemination time: the smaller the image, the lesser network traffic it will require. Moreover, the less time it takes for an image to disseminate, the earlier the nodes can activate energy saving features.

Another important aspect that rarely is taken into account while designing OAP systems is the efficiency of the integration between the OAP protocol and the underlying low-power Medium Access Control (MAC) layer, that is mandatory in long-running network applications. Anyway, it is straightforward to notice the contrasts there exist between requirements of a low-power MAC protocol and an OAP protocol. Indeed, to save power, a low-power MAC protocol increases the overall network latency which degrades the dissemination time of an OAP protocol. On the contrary, the high number of broadcast packets of an OAP protocol sensibly increases the duty-cycle of the MAC layer. In this paper we investigate this relevant aspect and show how, through the knowledge of the underlying MAC protocol, the efficiency of the OAP protocol can be significantly increased.

We used the official 2.1.2 release of the TinyOS operating system [52], which we ported on our hardware platform dedicated to WSNs, the MagoNode 1.1. TinyOS comes with an official OAP protocol named Deluge T2 and a low-power MAC protocol featuring Low Power Listening named BoX-MAC [83]. We made a preliminary experiment comparing Deluge with and without LPL showing the parameters we could leverage to increase its efficiency. We then proposed a simple reactive algorithm integrated in Deluge able to dynamically disable part of the LPL features to boost Deluge over-the-air reprogramming. With our simple approach, we are able to increase the performance of Deluge, compared to its standard implementation, by a factor of 2.6 in terms of energy efficiency and by a factor of 7 in terms of dissemination time.

In the remainder of this section we present the state of the art of OAP protocols (subsection 2.1.1) and we introduce BoX-MAC and Deluge (subsection 2.1.2). Furthermore, in subsection 2.1.3 and 2.1.4 we present the experimental data and analysis of the REACTIVE algorithm we implemented compared to other solutions. Finally, in subsection 2.1.5 we provide conclusions and possible directions for the future.

2.1.1 State of the Art

TinyOS does not support loadable modules, which means that the whole ROM image needs to be disseminated before reprogramming can occur. On the contrary, other operating systems, like Contiki [53], have the ability to load dynamic modules. These allow those systems to narrow propagation to updated modules only, thus, limiting the overall disseminated code size. In the last years similar techniques acting on the code structure in such a way that only the code differences are propagated and then reassembled locally on each mote before reprogramming, have been presented for TinyOS.

Zephyr. Zephyr [79] uses an optimized byte-level algorithm based on *rsync* to compute delta difference between ROM images. The computed delta is then disseminated and the ROM image reconstructed on each mote. The authors observe that computing byte-level delta is not enough, since small changes in the code might shift entire code blocks, thus, generating big sized deltas. To face this issue, the authors substitute function calls with jumps to fixed code locations acting as indirection table. So all the calls to a particular function in a user program will refer to the same entry in the indirection table. In this way if these calls are placed in different locations (i.e., shifted) only the value on the indirection table will be affected, thus keeping the delta between the two versions small.

Dynamic TinyOS. Dynamic TinyOS [80] has the ability to compile TinyOS components as separated ELF files that are linked together at run-time by a *TinyManager*. Separating the components in multiple files allows Dynamic TinyOS to disseminate only updated components while keeping the rest of the code unchanged. With Dynamic TinyOS it is also possible to group components whenever no updates are expected on those parts, in order to improve code optimization of the compiler. The result of this operation is a reduction in size of the ROM image.

Elon. Elon [81] introduces the concept of *replaceable* components. The principle relies on the fact that in TinyOS only few application-level components are expected to be updated while core modules are typically left unchanged. Hence, labeling few components as replaceable let the user to disseminate updates for those components only, keeping the update small. In addition, while the base version of replaceable components are stored in ROM, the updates are stored in RAM, without the need to use external flash.

Stream. Stream [82] stores in the external flash memory an image that implements the OAP protocol. When a network reprogramming is needed, Stream switches to the OAP image in order to disseminate the new application code. This technique allows Stream to keep a small overhead on the application code size, since it implements a very lightweight module whose job is to disseminate the command to switch to the OAP image. This small overhead greatly reduces the overall application code size, thus the dissemination time and network traffic. However, as pointed out in [81], in real-world complex applications, most of the core modules of TinyOS (i.e., radio driver, routing protocols, etc.) that are used by the OAP protocol, are already imported in the application itself, thus limiting the benefits of this approach.

We want to point out that most of the above solutions try to reduce the power consumption required by an OAP protocol by shrinking the size of the disseminated code. Those solutions are complementary to the one presented in this paper. Indeed, in our solution the power consumption is cut by boosting the Deluge dissemination time. Thus, reducing the code size, whenever is possible, further improves the performance of our approach.

2.1.2 BoX-MAC and Deluge

BoX-MAC [83] is the default low-power MAC protocol of the TinyOS operating system. It is based on the Low Power Listening technique where each node periodically wakes up and checks for incoming packets. Despite the period is fixed for all the nodes, the wake-up schedule is not shared. Hence, the transmitter has to repeatedly send the packet for the whole period length (figure 2.1). On unicast packets, the receiver notifies the transmitter with an acknowledgement message as soon as the packet is received, letting the transmitter to stop sending.

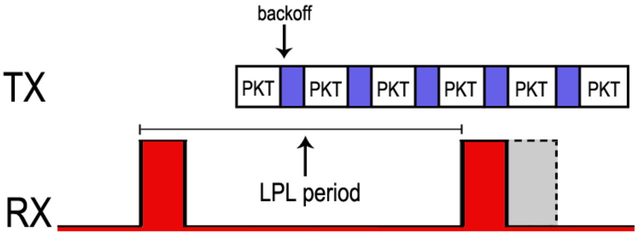


Figure 2.1: BoX-MAC transmission of a broadcast packet.

Despite Low Power Listening greatly reduces the radio power consumption, in [17] Langendoen et al. show how overhearing and collisions significantly degrade the energy efficiency of such approach. This happens especially under high traffic loads and in presence of broadcast packets. These conditions are exactly what we expect during network reprogramming: hundreds of packets, bearing the ROM image, are disseminated in broadcast over the network.

Thus, we have to keep into account the above cited issue when using OAP protocols on top of BoX-MAC.

TinyOS provides Deluge T2 [84], a reliable data dissemination protocol for propagating large data objects from one or more sources to all the other nodes of the wireless sensor network (WSN). Deluge can be exploited to support the over-the-air network reprogramming of a WSN, disseminating in a reliable way ROM images over the network. The start and stop of the network reprogramming is managed by the Drip [85] dissemination protocol, which disseminates broadcast messages carrying program version and commands. Drip is based on Trickle [86], a density-aware algorithm that organizes data dissemination in rounds. The round frequency decays exponentially to limit the number of packets retransmissions. However, when a message handling an updated command is detected, the frequency is restored to its initial value. Upon the reception of a Drip message (DRIP) carrying a *start command*, the mote starts transmitting broadcast advertisement messages (ADV) to notify the neighbors of its current version number and the portion of updated ROM it already has. The transmission of ADV messages is also based on the Trickle algorithm, thus, it is organized in rounds. At each round, all the motes check whenever there is a node with an updated version number and with updated portions of ROM. In such case, they perform an unicast requests (REQ) to their neighbor to start downloading data packets (DATA) containing the updated code. DATA messages are transmitted as broadcast packets to allow neighbors overhearing, thus, greatly reducing REQ and DATA messages.

2.1.3 Preliminary Experiments and Analysis

Setup. We analyzed the effects and interactions between the Deluge and BoX-MAC protocols, deploying a wireless sensor network in the basement of the the Department of Computer, Control and Management Engineering of Sapienza - University of Rome (figure 2.2).

The deployed sensor network is made of 10 MagoNodes 1.1 running TinyOS 2.1.2 and using the RFXLink library that implements BoX-MAC Low Power Listening. To emulate long-lasting network deployments, we set the duty cycle to 1%, which leads to a LPL period of 500 ms against a listen period of 5 ms. In addition, the application running on wireless motes is made of the Collection Tree Protocol (CTP) [56] which was programmed to periodically forward one dummy packet per minute toward a mote acting as sink. We enabled the Traffic Monitor provided by the RFXLink library [52] to keep trace of the radio power consumption, the number of packets transmitted and received. In addition, we used a modified version of Deluge that notifies the application layer the start/stop events related to the over-the-air reprogramming. When on of those events is triggered, a snapshot with the statistics is taken and

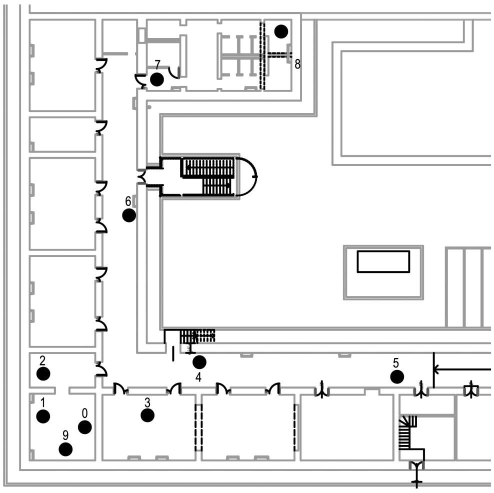


Figure 2.2: Testbed map at Department of Computer, Control and Management Engineering. Mote 0 is the sink node.

transmitted over the network exploiting the CTP protocol. In this way, we are able to take accurate measurements on ROM image dissemination time, power consumption and other statistics about motes. Finally, to better simulate a realistic scenario, radio modules operate with transmission power of about 7 dBm, which is lower than maximum value allowed by the hardware platform (+10 dBm).

Preliminary Experiments. The preliminary experiments consist in comparing the differences between wireless reprogramming of a fixed-size ROM image on a network running the Deluge protocol with LPL disabled, hence with 100% duty cycle, against the same job performed with LPL with duty cycle set at 1%. We will refer to these two modes as *NO-LPL* the former and *W-LPL* the latter. We use the following metrics to perform the comparison:

- *Dissemination time:* the average time elapsed from the start command, received by means of a drip message, to the completion of the image download

- *Radio duty cycle:* the average percentage of radio activity over the dissemination time

- *AVG TX packets:* the average number of packets transmitted by each mote

- *Packet types weight:* which represents the percentage composition of AVG TX packets in terms of DRIP, ADV, REQ and DATA packets.

We performed the preliminary experiment running 10 times the over-the-air reprogramming of a 28 KB ROM, which corresponds to the Blink application available in TinyOS 2.1.2 with Deluge support.

Preliminary Results. The results are summarized in table 2.1, figure 2.3a and figure 2.3b. The most relevant result is the sharp performance degradation in terms of dissemination time, which increases by a factor of 8 switching from NO-LPL to W-LPL mode, and the co-related increase of the AVG TX Packets by a factor of 100. The enormous increase in network traffic in the W-LPL mode increases the duty cycle up to 36%. This, combined with the longer dissemination time, makes the energy consumption to increase up to 3 times the NO-LPL one. Looking at AVG TX packets composition in figure 2.3a, we observe how in NO-LPL the main network traffic is made of DATA messages (81.2%) and, in smaller quantity, on ADV messages (13.6%). The Deluge description of subsection 2.1.2 explains these values: DATA packets represent the predominant part of the network traffic due to the multi-hop dissemination of the 28 KB ROM image. Furthermore, we recall that, when a new ROM block has been downloaded, the mote resets the Trickle timer to increase the ADV frequency with respect to its neighbors. This procedure, repeated several times on several nodes, makes the ADV packets to represent a significant portion of the overall traffic.

Table 2.1: Preliminaty results

	NO-LPL		W-LPL	
	Value	Std-dev	Value	Std-dev
Time (s)	65	12.7	524	27.6
Duty Cycle (%)	100	0	36	10
Energy (J)	3.29	0.3	9.6	3

Figure 2.3b shows the sharp increase of the transmission network traffic when enabling W-LPL. The increase is explained by the fact that each packet transmission in W-LPL is repeated hundreds of time, as long as the 500 ms LPL period. Despite an increase was expected, what is noticeable is the difference of the packet types weight: while with NO-LPL the DATA packets were representing more than 80% of the overall traffic, in W-LPL it reduces to 42.2% only, while ADV messages become the predominant part of the network traffic, growing from 13.6% with NO-LPL to 52.9% with W-LPL. To find the reason for this unexpected behavior we looked at the events that trigger a reset of the Trickle timer for ADV dissemination. These events are:

- a new block has been stored;

- the whole image has been downloaded;
- reception of the Drip start command;
- reception of an ADV from a neighbor node that has different number of blocks.

While the behavior caused by the first three events are almost the same in both W-LPL and NO-LPL mode, the response at the occurrence of the latter is different. Indeed, using LPL increases the probability that the nodes have different block numbers during the dissemination phase. This implies that ADV received from a neighbor will trigger a reset on the Trickle time, increasing the transmission rate of ADV messages.

From this preliminary experiment we can conclude that introducing LPL drastically degrades the overall OAP performance in terms of dissemination and power efficiency, as expected. On the other side, LPL must be enabled when running long-lasting sensor networks.

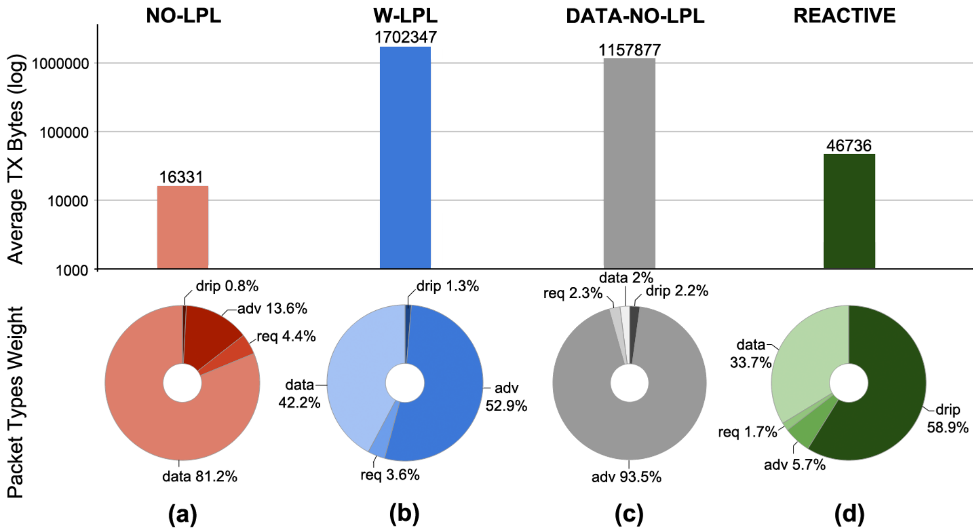


Figure 2.3: Network traffic during over-the-air reprogramming

2.1.4 Proposed Algorithms

DATA-NO-LPL. An easy and intuitive method, inspired by Stream (subsection 2.1.1), to mitigate the effect of LPL on Deluge performance can be summarized as follows: upon the reception of the start command, switch from the current running application to a dedicated *golden image* already stored in ROM which does not use LPL. The only goal of the golden image is to let a new image version get disseminated over the wireless network. At the end of the dissemination, a node switch to the new image version which uses LPL. The drawback of this approach is that every node in the network must receive the dissemination command almost at the same time. In this way all nodes are able to participate to the dissemination phase, switching to the golden image and so disabling LPL. Otherwise, nodes are unable to receive packets of the new image version since they are still using an image featuring LPL, instead of the golden image.

Recalling the outcomes of subsection 2.1.3, we conceived a new approach, named DATA-NO-LPL, that drastically reduces the network traffic by dynamically disabling LPL from DATA packets. The proposed solution is able to guarantee state correctness, i.e., the motes consistently switch between LPL and non-LPL modes, and, thanks to the sensibly lower network traffic, greatly reduces the image dissemination time.

To describe in details our approach, we first need to introduce the following definitions: we refer to *full-LPL* as a mote running LPL both in transmission and reception, while we will refer to *tx-LPL* as a mote running LPL in transmission while keeping a fully-on radio in reception. First of all, we observe that a mote running tx-LPL does not compromise the overall mote-to-mote communication since an always-on receiver is still able to receive packets from an LPL transmitter. Symmetrically, a tx-LPL mote would continue to transmit using LPL, which would let the mote to communicate with another mote running full-LPL. As a matter of fact, a hybrid network of motes running full-LPL and tx-LPL would not compromise the overall network functionality.

DATA-NO-LPL takes advantage of this observation and uses the Drip messages, that carry the start/stop commands of Deluge, to trigger a switch between full-LPL and tx-LPL on all the motes upon the reception of a start command. A mote running our algorithm would switch back to full-LPL at the completion of the new image download. This simple algorithm allows the mote to react dynamically to the start of an over-the-air reprogramming without the need to add any kind of control overhead to the protocol. Of course this is not enough to solve the performance degradation observed in the previous experiments, since it would not disable the LPL in transmission, which is the main responsible of the performance degradation. Moreover, keeping the radio always on increases the energy consumption. However, this simple algorithm allows us to perform some optimization.

In particular, DATA-NO-LPL is based on the observation that DATA messages in Deluge are transmitted upon the reception of a REQ message. This means that the mote sending the REQ message has already received the start command from Drip, thus it is running tx-LPL. This implies that the mote is able to receive packets without LPL, so we can safely disable LPL transmission on the DATA packets. Since we have seen that DATA packets represent half of the overall traffic in an OAP protocol running LPL, without LPL we should notice a great reductions in dissemination time. This lead us to repeat the same test performed in the preliminary experiments running Deluge with DATA-NO-LPL. Results are shown in table 2.2 and figure 2.3c. We figure out that, as expected, the dissemination time drops up to 50% when compared with the W-LPL mode and that the DATA messages represent 42% of the overall traffic when transmitted in W-LPL mode, while in NO-LPL mode they represent only the 2%.

Despite the great reduction in both dissemination time and transmitted DATA packets, the overall energy consumption increases from 9.6 Joules of the W-LPL mode to 14.9 Joules of the DATA-NO-LPL one. This sensible increase is related to the fact that the 50% reduction in dissemination time is compensated by fact that the radio duty cycle is 100% in DATA-NO-LPL, while in W-LPL it is 36%. In order to outperform the energy consumption of the W-LPL mode, our solution should reduce the dissemination time up to 70%.

REACTIVE. Looking back to figure 2.3c, we notice that most of the remaining network traffic is now related to ADV messages (93.5%) that are still transmitted in LPL. This is the reason why REACTIVE, which extends DATA-NO-LPL, leverage on the ability to dynamically disable LPL on ADV messages. As opposed to DATA-NO-LPL, removing LPL from ADV messages is not straightforward. Indeed, issues arise during start and stop transition phases, where not all the nodes share the same LPL state, in particular:

- when starting over-the-air programming, nodes can be in tx-LPL or full-LPL based on the fact they already received the Drip start message or not;
- when finishing the download of the image, the node switch back to full-LPL but other motes might still be downloading the image, hence, they might still be in tx-LPL

The problems during transition phases is that an ADV message transmitted without LPL could not be overheard by its neighbor. Actually, the start phase transition does not represent an obstacle keeping state correctness in the network since a node that has not yet received the Drip start message, would anyway ignore incoming ADV messages. On the other side, as soon as it receives the start message, it would switch to tx-LPL, thus receiving ADV messages sent without LPL. The real obstacle is represented by the early switch during the stop transition. In fact, a node switching back to full-LPL will no longer provide DATA packets to a node in tx-LPL. This will prevent the him to finish the image download, keeping it in tx-LPL almost forever.

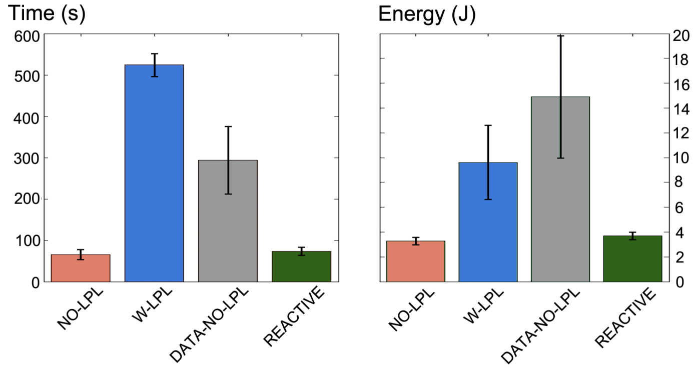


Figure 2.4: Performance summary in terms of dissemination time and energy consumption

To prevent this issue, REACTIVE uses a timeout timer initialized to a value τ that determines whether the ADV messages are sent in LPL or not. The principle is that if a mote finishes the image download, it is prevented from switching back to full-LPL until the timer fires. However, the timer is reset each time the mote detects an ongoing reprogramming activity. This means that the timer is reset when:

- the mote sent a REQ message;
- the mote sent a DATA message;
- the mote receives a REQ message;
- the mote receives an ADV message from a mote that requires data.

This ensures that the mote does not switch back to full-LPL until reprogramming activity is detected in the surrounding neighborhood. It is important to point out that this simple timeout timer added to our algorithm also guarantees state correctness even in special cases where the timeout timer erroneously elapses. In fact, let us consider two nodes that are performing over-the-air reprogramming. The one with the less up-to-date ROM image is requesting data from its neighbor. But if the neighbor has already finished updating its ROM image and if the τ value of the timeout timer is too short, it will switch back to full-LPL. This will leave the former node in tx-LPL without the possibility to communicate with its neighbor. However, the absence of communication will prevent the timeout timer to reset, thus, at some point it will fire on this node too. As a result, the node will start sending ADV messages with LPL enabled that will notify its neighbor and resume the communication.

We tested REACTIVE following the same experimental setup described in subsection 2.1.3, setting the τ value equal to 4 seconds. The results are summarized in table 2.2 and figure 2.3d. Table 2.2 shows how REACTIVE shrinks the dissemination time up to 7 times when compared to W-LPL and up to 4 times when compared to DATA-NO-LPL. Motes in REACTIVE have an average energy consumption of 3.7 Joules which is 61.5% less than W-LPL. Indeed REACTIVE dynamically disables most of LPL communication during an over-the-air programming phase. This brings REACTIVE performance close to NO-LPL, but with the advantage of running on a long lasting wireless network based on low power MAC protocols such as BoX-MAC. Figure 2.4 shows a summary of the results observed in this paper.

Table 2.2: Algorithms results

	DATA-NO-LPL		REACTIVE	
	Value	Std-dev	Value	Std-dev
Time (s)	294	82	73	10
Duty Cycle (%)	100	0	100	0
Energy (J)	14.9	4.9	3.7	0.3

2.1.5 Concluding Remarks

In this section we presented a reactive approach to the Over the Air Programming (OAP) for Wireless Sensor Network in a LPL scenario. Our modified protocol, named REACTIVE, minimizes the reprogramming overhead of LPL, by dynamically shrinking LPL transmissions of Deluge T2 protocol. In this way we can obtain much better performance than the Deluge standard version running on LPL. Indeed, our experiments show that the image dissemination time is 7 times smaller than Deluge with LPL, while the energy consumption falls 2.6 times. The advantage of our solution is that it perfectly integrates in Deluge with LPL and can be complementary to other solutions seen in subsection 2.1.1 that are focused on image size reduction.

2.2 Wake-Up Based Routing Protocols

Duty cycling the mote main transceiver (i.e. allowing a WSN node to turn on and off its radio apparatus according to a pre-set period) allows to save energy and so to achieve longer networks lifetimes. However, duty cycling technique do not eliminate completely the idle listening of the main transceiver, and comes at the price of high latency.

For this reason, research has recently shifted towards the design of communication architectures for WSNs based on wake-up radios (see section 1.2), i.e., ancillary transceivers used for the sole purpose of waking up neighboring nodes, and whose power consumption is considerably lower than that of the main radio [87]. The aim is to eliminate the need for duty cycling, and therefore idle power consumption, while keeping latency at bay. The sole capability of waking up neighbors, however, would not provide the major improvements required by new critical applications, as nodes not interested in communication will be up wasting energy in overhearing and contention. Wake-up architectures are therefore being defined to allow nodes to wake up specific neighbors for communication (selective awakenings). This fundamental capability, however, brings forward new requirements and challenges for designing selective awakening-aware protocol stacks that take also into account the limited range and low data rates typical of the wake-up radios.

In this section we contribute to the research on wake-up radio enabled sensing systems by proposing and evaluating ALBA-WUR, a cross-layer protocol for data gathering in wireless sensing systems that redesigns ALBA-R protocol [29]. In particular, we extend ALBA-R it to exploit the features of the wake-up radio hardware platform we introduced in section 1.2. We decided to enhance ALBA-R capabilities with the wake-Up radio technology, since this protocol is particularly suitable to demonstrate the flexible addressing capabilities of our WRx. Indeed, ALBA-R is a protocol where next hop relays are chosen based on an array of different policies. These include the current traffic and channel conditions, and the geographic advancement towards the data final destination (sink) that the neighbors can provide. With ALBA-WUR we show that the complex relay selection policies used by ALBA-R can be naturally expressed by one or more semantic addresses signifying the current status of a node, which the node itself can locally compute and take over time. A sender can therefore selectively wake-up only those neighbors whose semantic addresses match the policy, thus engaging in communications only nodes whose current status makes them the best relays. This idea, much more than ID-based wake-up, enables greater freedom in protocol definition and leads to significantly better network performance.

We compare the performance of ALBA-WUR and ALBA-R via simulations. In particular, we implemented both protocols in GreenCastalia [88], a recently proposed open-source extension of the Castalia simulator [89] that focuses on modelling energy-related aspects of WSNs. In our experiments, nodes running ALBA-R follow a nominal duty cycle d, with $d \in \{0.01, 0.03, 0.1, 1\}$, i.e., they stay awake only 1%, 3%, 10% of the time, and all the time, respectively. Nodes running ALBA-WUR are endowed with a wake-up receiver modeled after our WRx (see section 1.2.2), with communication parameters set to values from measurements on real prototypes. Performance results in WSN scenarios with increasing traffic show impressive performance gains of ALBA-WUR over ALBA-R. In particular, nodes running ALBA-WUR in typical WSN application scenarios are able to delivery data to the sink by spending 300 times less energy than nodes featuring the same harware using duty cycle, even if the duty cycle is very low. With respect to ALBA-R, in networks with longer duty cycles ($d \geqslant 0.1$), ALBA-WUR achieves even greater improvements, obtaining energy savings ranging from two to five orders of magnitude. This translates in estimated network lifetimes that are decades longer than those achievable in networks with duty cycles. Results concerning latency show that the use of WRx technology is effective in reducing the time needed to deliver packets to the sink. In the specific, the end-to-end latency of packets routed by ALBA-WUR is quite close to that incurred by packets routed by ALBA-R with no duty cycle, and well below that of packets routed by ALBA-R with $d = 0.1$. These results are possible because of the unique combination of our cutting-edge wake-up radio technology with ALBA-WUR, where the very effective relay selection policies of ALBA-R are re-written into semantic addresses for WRx-based selective awakenings.

The remainder of this section is organized as follows. First, we introduce the state of the art of radio wake-up assisted protocols (subsection 2.2.1). Then, we describe in details ALBA-R and ALBA-WUR protocols (subsection 2.2.2). Finally, in subsection 2.2.3, we present performance evaluation results

2.2.1 State of the Art

Only a handful of existing approaches on protocol design for WSNs explicitly consider the presence of a wake-up device. The work presented by Stathopoulos et al. [90] considers a dual radio system in which the low power radio is used as a multi-hop control channel in charge of determining an end-to-end route for data packets. Once the route has been received (from a "topology controller" node) on the low-power radio, it is used for data forwarding on the main radio by selectively waking up nodes along the route. Addressing of nodes to be woken up is still ID-based, i.e., it is not dynamic or dependent of network/node status. Latency is also a main issue in this case, as route discovery on a low data rate network is time consuming.

Marinkovic et al. present a protocol for single-hop Wireless Body Area

Networks (WBANs) that works with a wake-up receiver presented in [91]. The protocol is based on a TDMA scheme where slots are assigned to the nodes by a master node (MN). Slots are dynamically assigned by the MN, which wakes up sensors with specific functions, as needed. Wake-up addressing is solely based on the node IDs. While this protocol works well in a small, single-hop network like a WBAN, it lacks in flexibility for more general WSNs applications.

To the best of our knowledge, Blanckenstein et al. present the first work where wake-up addressing is dynamically used to wake-up nodes depending on the data they have sensed [92]. In particular, a protocol is defined to cluster nodes with similar data readings. Wake-up packets are used to disseminate similarity information about the sensor readings and a cluster head is in charge to forward one copy of the data. The large space of possible readings is factored down to the limited space of wake-up patterns, that the nodes dynamically assume as addresses, thus being able to participate in clustering. The paper models a wake-up receiver by Hambeck et al. [93] taking into accounts missed detection and false wake-up rates. The wake-up receiver considered for this work has energy consumption higher than that of our WRx (see section 1.3.1), and considerably larger wake-up times, which have a significant impact on the overall energy consumption and on data latency.

The beneficial effects of a novel wake-up receiver on WSN performance is shown by Petrioli et al. in [94]. The presented wake-up receiver operates in the 2.4 GHz band and is capable of selective addressing. This feature is demonstrated on protocols for interest dissemination and for routing information suitably designed to work with the receiver. While operations in the 2,4GHz have the advantage of allowing main and wake-up transceivers to share a single antenna, the proposed technology is quite energy expensive, in that wake-up operations are in the order of mW, three orders of magnitude higher than in our case (see section 1.3.1).

2.2.2 ALBA-WUR: Extending ALBA-R with Wake-Up Radio

ALBA-R [29] is a cross layer geographic protocol for convergecasting in WSNs. It features the integration of awake/asleep schedules, MAC layer, routing protocol, load balancing, and back-to-back packet transmissions (ALBA) with a mechanism for recognizing and routing around connectivity holes (Rainbow).

ALBA. The basic operation of ALBA is as follows. Nodes in the network alternate between awake/asleep modes according to independent wake-up schedules with fixed duty cycle. When a node x has packets to forward, it polls its neighbors by broadcasting a Request-to-Send (RTS) packet. Neighboring nodes that are currently awake respond with a Clear-to-Send (CTS) packet carrying information based on which x can choose the best relay. More specifically, for each neighbor y that has sent a CTS packet, node x computes the following two parameters: the *queue priority index* of y (QPI_y), and the *geographic priority index* of y (GPI_y). The parameter QPI_y quantifies how fast and well node y can move packets forward, taking into account its recent past performance as a relay. It is defined as $\lceil (Q_y + N_B)/M_y \rceil$, where Q_y indicates the number of packets in the queue of node y, N_B is the number of packets that node x wants to transmit in a burst (back-to-back transmissions, up to N_B^{max} packets), and M_y indicates the number of packets that node y was able to transmit back-to-back, without errors, in the recent past (it is a moving weighted average of recent transmission attempts). The QPI has been designed so that if node y is congested (i.e., has a high queue occupancy Q_y) or it has been a "bad forwarder" (experiencing high packet transmission errors, i.e., it has a low M_y) it is less frequently chosen as a relay.The selection of relays with low QPI therefore aims at decreasing latency at each hop by balancing the network load among good forwarders. The parameter GPI_y indicates the geographic advancement towards the sink that node y can provide. It is computed based on the locations of node x, node y and of the sink. The GPI is used to discriminate among relays that have the same forwarding performance, i.e., the same QPI. Nodes with higher GPI are preferred.

Rainbow. GPI-based forwarding implements the typical greedy approach to geographical routing that works well whenever relays towards the sink can be found, e.g., in dense networks. In sparser WSNs, or in networks deployed around obstacles, greedily choosing the next hop relay is not always possible due to the presence of dead ends, which hinders routing performance severely. Rainbow extends ALBA to provide a mechanism for detecting and routing around connectivity holes whenever they occur in WSNs. The basic idea is allowing the nodes to forward packets away from the sink when a relay in the direction of the sink cannot be found. In order to remember whether to seek for relays in the direction of the sink (positive advancement area F) or in the opposite direction (negative advancement area F^C) each node is labeled by a color (from a given list) and seeks for relays among neighbors with its own color or with the color immediately before in the list. Hop by hop forwarding follows the rules established by ALBA. As an example, assume the following list of possible colors: yellow, red, blue and purple. Initially, all nodes are colored yellow, and function according to the basic ALBA rules, i.e., they look for relays located in the region F that provides positive advancement towards the sink. If no connectivity holes are encountered, all nodes remain yellow, keeping finding relays in the direction of the sink. Nodes that cannot find relays offering positive advancement infer that they may actually be dead ends, and change their color to red. According to Rainbow, red nodes will send the packet away from the sink by searching for yellow or red nodes in their region F^C. If a red node cannot find red or yellow nodes in F^C, it changes its color again, becoming a blue node. It will now look only for blue or red relays in F. Similarly, a blue node that cannot find blue or red relays in F turns purple, and starts looking for purple or blue nodes in F^C. This process continues until all nodes have converged to their final color and allows nodes to discover routes to the sink traversing the minimum number of colors, independently of localization errors and connectivity.

ALBA-WUR. We have redesigned ALBA-R to work with the wake-up features of our WRx. When a node x has packets to transmit, it wakes up only those neighbors that satisfy the following conditions. To start with, only nodes with the appropriate color will be woken up. For instance, if x is blue, the wake-up address should be chosen so that only nodes that are either blue or red will wake-up. Within this set of nodes, node x will want to wake up those nodes y that can receive the number of packets N_B that x needs to transmit, and with their M_y above a given threshold. To do so, x broadcasts a wake-up packet made up of the following fields: its color, the number of packets N_B and the target QPI it is currently looking for. Figure 2.5 shows the format of the wake-up packets.

Format of an ALBA-WUR WRx packet

Preamble	Sender Color	Burst Size (N_B)	Target QPI

Format of an ALBA-WUR WRx address

Acceptable Color	Acceptable Burst	QPI

Figure 2.5: Structure of an ALBA-WUR packet.

Nodes that can serve as relays indicate their availability by assigning to themselves a pool of WRx addresses, dynamically updated based on their status, each of which expresses a combination of the three values: sender color, burst size and target QPI the node can satisfy. For example, according to Rainbow, a node whose color is blue can only serve as relay for nodes with color blue or red. Thus, the pool of its WRx addresses will contain only sequences with either Color = blue or Color = red (see example below). The last two fields of the WRx address are set based on the queue occupancy status of the node and on its forwarding history. In particular, for each possible burst size that the node can accept based on its queue status, it computes the corresponding QPI value and stores it in the last field of the address. When a wake-up packet is received, addresses in the pool are checked for a match. If the packet matches one of the addresses, i.e. if node actual resources are compliant with sender constraints, the node wakes up. Figure 2.5 shows the format of a wake-up address. The addressing mechanism of ALBA-WUR is illustrated by the following example. Let us consider a red node y that currently has one packet stored in its queue, i.e., $Q_y = 1$, and such that its moving weighted average of recent transmission attempts is 2, i.e., $M_y = 2$. If the maximum number of packets that can be sent in a burst is $N_B^{max} = 2$, to serve as relay node y will have four wake-up addresses in its pool, each corresponding to a combination of color (red or blue), burst size and the target QPI it can satisfy (see table 2.2.2). QPI values are computed as detailed above for ALBA-R, i.e., $\lceil (Q_y + N_B)/M_y \rceil$.

Color = red	Burst size = 1	QPI = 1
Color = red	Burst size = 2	QPI = 2
Color = blue	Burst size = 1	QPI = 1
Color = blue	Burst size = 2	QPI = 2

A node x, whose color is blue, with $N_B = 1$ data packets to transmit and looking for a relay with the best possible QPI, will broadcast a wake-up packet with the address in table 2.2.2.

Color = blue	Burst size = 1	Target QPI = 1

The neighbors of node x that receive this packet will wake up their main microcontroller and radio only if the wake-up packet matches one of the addresses in their pool. For instance, node y would wake up, as the third address in its pool matches the wake-up packet sent by x. After the selective awakening phase, node x waits for neighboring nodes to turn their radio on, and sends an RTS packet using the main radio. The next-hop relay is then chosen according to ALBA-R operations. If no neighbor is found that can satisfy the requests of node x, the awakening process is repeated with x now lowering its demands, i.e., seeking for nodes with different status (e.g., a lower QPI and/or burst size).

It is worth noting how the complex relay selection policies that make ALBA-R one of the high-performance protocols for WSN routing are naturally expressed by the dynamic semantic addressing of ALBA-WUR, as simply implemented via our WRx technology. Protocol design is no longer driven by the need of trading off energy consumption and lifetime for latency, as efficiently implemented semantic addressing enables senders to quickly select only those among their neighbors that can best further the routing process. In addition, if address matching is performed via software by the optional on-board microcontroller, further optimization is possible that can adapt the matching to specific deployment conditions. For instance, in noisy environments in which the WRx is subject to interferences that can result in false negatives, an error-tolerant or partial address matching mechanism can be employed. Furthermore, in case of a large number of wake-up addresses, the address pool can be stored in a space-efficient data structure, such as a Bloom filter.

2.2.3 Simulation Results

We compare the performance of ALBA-WUR and ALBA-R via simulations. We implemented both protocols in GreenCastalia [88], an open-source extension of Castalia [89].

Node modeling. We extended GreenCastalia to model nodes with multiple transceivers: The Atmel Atmega128RFA1 System-On-Chip of the MagoNode, and the TI CC1101 used by the WTx module to transmit wake-up sequences. For the main radio we use the default settings of GreenCastalia for channel and radio models. The average path loss between nodes in the network follows the lognormal shadowing model. Packet reception probability for each link is computed based on Signal to Interference plus Noise Ratio (SINR), packet size, and modulation type.

We implemented a GreenCastalia module to model our WRx including also a PIC12LF1552 as on board optional microcontroller dedicated to address matching (see section 1.2.2 for further details of our WRx module). The WRx is always kept in idle listening with the PIC12LF1552 in sleep state, consuming an average current of 600 nA.

A novel energy model is also defined based on which nodes equipped with the WUR system (i.e. both WRx and WTx modules) operate in one of five power modes, depending on the state of their microcontroller and transceivers (table 2.3). To accurately characterize the energy model we experimentally measured the power consumption of the node in each state, as well as the transition time between two states, and the average power consumption of each node component during state switching.

Table 2.3: Power modes of a node equipped with the WUR system

Power Mode	Main Radio	MCU	WTX	WRX
TX	TX@-2dBm	Active	Power Down	Idle List.
RX	RX	Active	Power Down	Idle List.
WUR-TX	Power down	Active	TX	Idle List.
WUR-RX	Power down	Power Down	Power Down	RX
SLEEP	Power down	Power Down	Power Down	Idle List.

We model the wake-up probability as a function of the power received by the WRx and of the data rate used to transmit wake-up sequences (figure 2.6).

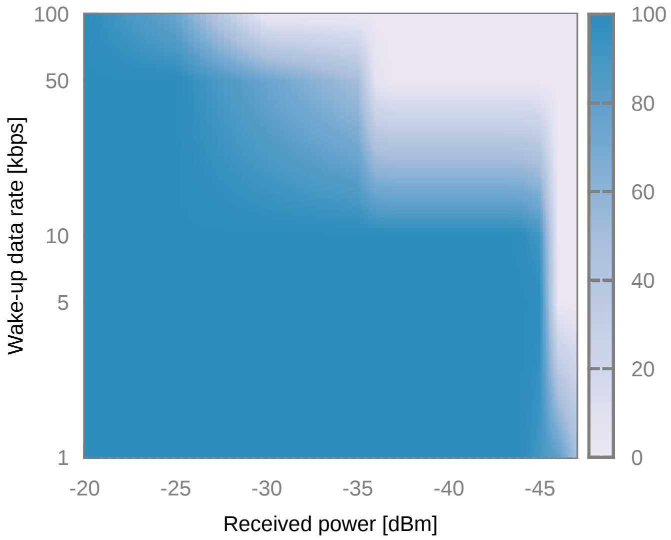


Figure 2.6: Wake-up probability vs received power

The additive interference model is used for both main radio and WRx, so that the effect of simultaneous transmissions from multiple nodes is linearly added at the receiver.

Scenarios and Settings. We consider networks where 120 nodes are randomly and uniformly distributed over an area of 200 m×200 m. The sink is also randomly placed in the area. The transmission power of a node main transceiver is set to -2 dBm and the data rate is fixed to 250 kbps. In order to obtain longer routes, and create connectivity holes that would challenge geographic forwarding, a rectangular obstacle is placed within the deployment area. This results in topologies with an average of three to four colors. In line with experimental measurements (see section 1.3.2), the sensitivity of the WRx is set to -50 dBm. The size of a wake-up packet is 8 bits for coding color, burst length and QPI, preceded by a preamble of 8 bits. We set the data rate of the WTx module to 1 kbps and 5 kbps. Data traffic is generated according to a Poisson process of intensity λ packets per second. Once generated, each data packet is assigned to a source node chosen randomly and uniformly among the network nodes, except the neighbors of the sink that only act as relays. The size of each data packet is 128 B. Nodes have queues capable of storing a maximum of 20 packets. ALBA-R parameters are set as follows. The node asynchronous wake-up schedule follows a fixed duty cycle

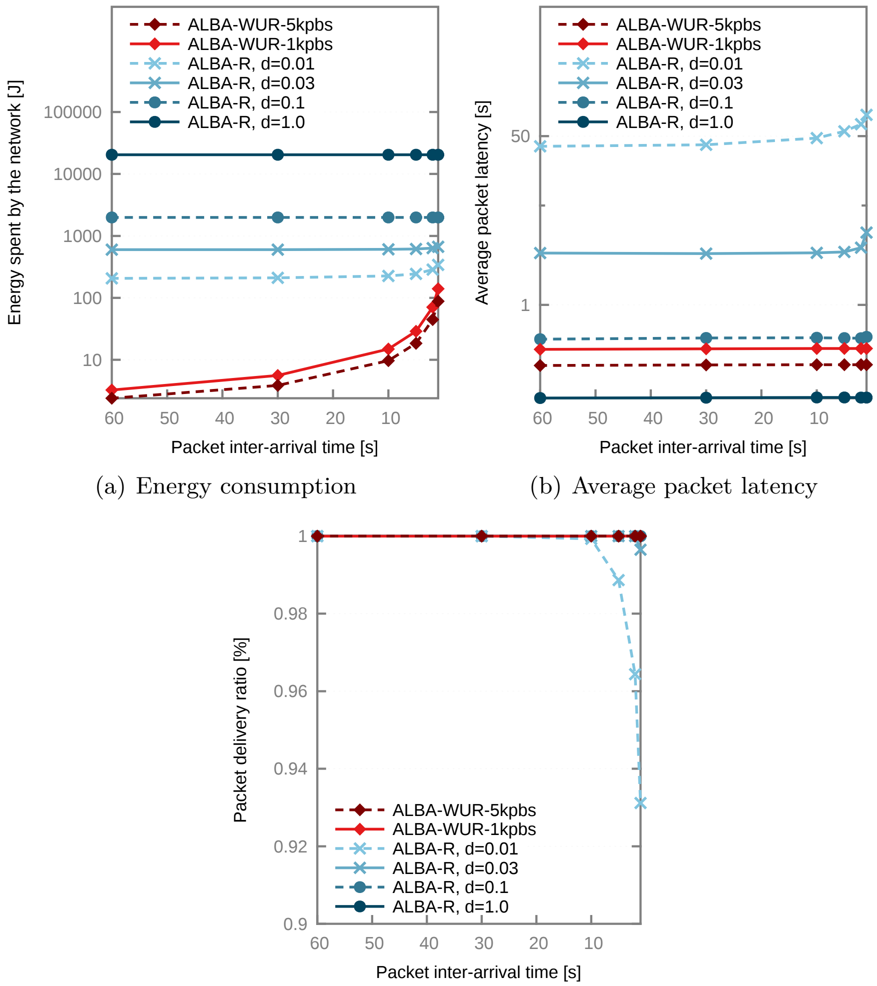
$d \in \{0.01, 0.03, 0.1, 1\}$. The maximum QPI and burst size N_B^{max} are both set to 5. While simulating ALBA-R, the GreenCastalia modules implementing the WRx and the wake-up transmitter are not instantiated and have no impact on energy consumption. All results have been obtained by averaging the outcomes of a number of simulation runs large enough to obtain a 95% confidence interval and 5% precision. Each run lasts 3600 s. In order to evaluate steady-state performance, all metrics are collected after each node has acquired its final color.

Performance Metrics and Results. We evaluate the performance of ALBA-WUR and ALBA-R with respect to the following metrics:

- the total energy consumed by the network (but the sink);
- the average latency, computed as the mean of the end-to-end latency of each packet successfully delivered to the sink;
- the packet delivery ratio, i.e. the fraction of packets that are successfully delivered to the sink;
- the expected lifetime of the network, defined as time when the first node dies.

Results on the performance of ALBA-WUR and ALBA-R are shown in figure 2.7 and figure 2.8. In particular, we show results for different variants of the two protocols, namely, ALBA-WUR with two different wake-up data rates, and ALBA-R with four different duty cycles.

Energy consumption. Figure 2.7(a) depicts the energy consumed by ALBA-WUR and ALBA-R. We observe that in typical WSN scenarios, with relatively low traffic, where protocols such as ALBA-R would efficiently operate with very low duty cycles (e.g., $d = 0.01$ or $d = 0.03$), ALBA-WUR obtains remarkable improvements on energy consumption, spending up to 100 times and 300 times less energy, respectively, than ALBA-R: a change driver for a wide variety of WSN applications. Results are even more remarkable for higher values of the ALBA-R duty cycle ($d \geqslant 0.1$), where we can observe improvements from two to five orders of magnitude. The performance of ALBA-WUR depends on the wake-up data rate. Sending wake-up packets at 5 kbps reduces energy consumption between 26% (low traffic) and 37% (high traffic) with respect to when those packets are sent at 1 kbps. This result may seems counterintuitive since using lower wake-up data rates implies lower energy consumption because of the higher wake-up probability (figure 2.6), which reduces the number of attempts needed to successfully wake up a node. However, this beneficial effect is limited by the increased amount of energy needed to transmit wake-up



(a) Energy consumption

(b) Average packet latency

(c) Packet delivery ratio

Figure 2.7: ALBA-WUR vs. ALBA-R for increasing traffic.

packets. Indeed, in order to send a packet at 1 kbps the transceiver need to stay on for an amount of time five time higher than the case in which the data rate is 5 kbps. We notice that with increasing traffic the energy consumption of ALBA-WUR also increases.

Latency. The average end-to-end latency of packets delivered using ALBA-WUR and ALBA-R is shown in figure 2.7(b). The performance of ALBA-WUR depends on the data rate used to transmit wake-up packets. More specifically, using a wake-up data rate of 5 kbps results in latencies about 32% less that those incurred transmitting wake-up packets at 1 kbps. Predictably, ALBA-R with no duty cycle achieves the best performance, incurring a latency of 115 ms. This latency, however, is only half of that of ALBA-WUR with a wake-up data rate of 5 kpbs. In many application scenarios, this is quite a small price to pay to obtain a five orders of magnitude improvement in energy consumption (figure 2.7(a)). Duty-cycling, as expected, significantly affects latency: when $d = 0.1$, the performance of ALBA-R is almost two times better than that of ALBA-WUR, despite the average length of routes found by ALBA-WUR is twice that of the routes found by ALBA-R (because of the different transmission ranges of the WRx and of the main radio). Using lower duty cycles makes it difficult for ALBA-R nodes to find neighbors that are awake, which results in an average increase of the end-to-end latency of up two orders of magnitude with respect to that of ALBA-WUR. For example, ALBA-R with a 3% duty cycle exhibits an average latency of 3.75s, which makes it unfit for time-critical WSN applications. This value increases to several tens of seconds for the smallest value of the duty cycle we considered here ($d = 0.01$).

Packet delivery ratio. Figure 2.7(c) shows the packet delivery ratio of ALBA-WUR and ALBA-R for increasing traffic. Independently of traffic, ALBA-WUR consistently achieves a packet delivery ratio of 100%. ALBA-R with duty cycle higher than 3% successfully delivers all data packets to the sink. The packet delivery ratio of ALBA-R operating with $d = 0.3$ is always higher than 99.9%. When using the lowest duty cycle, however, the performance of ALBA-R becomes sensitive to increasing traffic. In particular, the percentage of packets delivered by ALBA-R when $d = 0.01$ is $< 99\%$ for a packet inter-arrival time of 5s, and falls below 93% when packets are generated to the network once per second. This is due to the difficulty of finding a relay, which results in large numbers of packets being dropped because the queues are full.

Network lifetime. The final set of experiments concerns the expected network lifetime of a WSN running ALBA-WUR or ALBA-R for increasing traffic. We perform a conservative estimation of the network lifetime by computing the expected time at which the first node in the network depletes its energy. In our calculation we assume that the traffic patterns observed during our one hour simulations are representative of the traffic at each node in the long term. Nodes are powered by two 1.5V AA alkaline batteries whose capacity is 2500 mAh [95]. Results are shown in figure 2.8, which depicts the estimated number of years for which a network using ALBA-WUR or ALBA-R is expected to operate under different traffic generation rates. The remarkably low energy consumption of ALBA-WUR (figure2.7(a)) results in expected network lifetimes unheard of for WSNs. Under light traffic conditions, i.e. when one packet per minute is generated to the network, ALBA-WUR obtains an expected network lifetime of several decades (if only batteries could last that

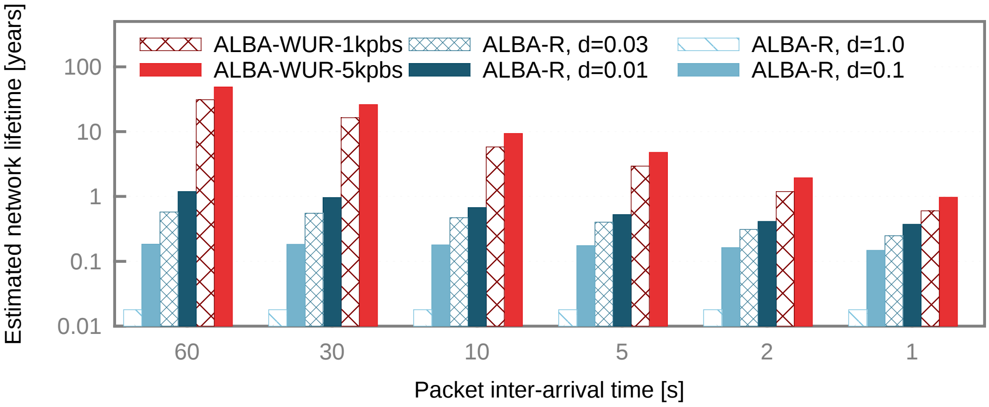


Figure 2.8: Network lifetime (the y axis is logscale).

long). This is an impressive improvement with respect to the duration achievable by a WSN with duty cycle: a network with nodes sporting two 1.5V AA alkaline batteries, strictly following a nominal duty cycle of 1% and operating in a scenario with no traffic, would last less than 2 years. As explained while discussing figure 2.7(a), when traffic increases the ability to save energy, therefore increasing the network lifetime, is constrained by the need to often keep nodes on to transmit and receive data packets. This motivates why differences in terms of network lifetime reduce at higher inter-arrival times. Nevertheless, for a packet inter-arrival time of 2s (1s) ALBA-WUR is still able to extend the network lifetime by a factor 5 (3) with respect to ALBA-R with 1% duty cycle.

Chapter 3

Energy Neutrality

In recent years the need for WSNs able to operate for an amount of time in the order of years emerged. Fulfilling the needs of this particular application requires new energy sources to replace batteries or to be used alongside them. This alternative is provided by the energy coming from renewable sources such as solar, wind, thermoelectric, etc. This energy is available in the ambient surrounding motes and it has to be harvested and accumulated using proper hardware components. A platform able to do that it is called energy harvesting platform. An energy harvesting platform exploiting an energy scavenging mechanism is able to recharge the batteries of motes increasing their lifetime. This kind of platforms have been widely studied by researches in the recent years [11, 21–24].

The naïve way of using harvested energy is to treat it as a supplement of battery energy, trying to maximize the lifetime of motes through energy management. A different approach is possible: exploit harvested energy with a suitable rate, allowing motes to operate perennially. This mode of operation is called energy-neutral (EN) mode [12]. Using energy neutrality definition we can say that a mote, geared with an harvesting module, reaches energy-neutral state if it can support a certain level of workload forever. In this context, an energy neutral mote it is not just a WSN mote interfaced with an energy harvesting module, but it is an embedded system aware of the amount of energy available in the surrounding environment and capable of predicting the quantity of energy that it will be able to acquire in the next future. These abilities allow it to decide which will be next activities it will accomplish (i.e. sense, transmit data, go to sleep state) basing on the acquired information. This requires ad hoc design of the software in order to use in the best way the availability of energy in the surrounding environment, which could change over space and time and in relation to tasks that a mote has to complete. For example, using photovoltaic panels, the energy changes over time, between day and night, and space, between zones in shadows and zones in full sun.

Since the energy-neutral approach is so different from lifetime maximization approach typical of WSNs, power management design observations made for the latter do not hold for the former. This opened a new research branch in WSNs and rises at least two fundamental questions (posed in [12]):

1 *Energy-Neutral Operation (ENO):* How a mote can operate in such a way that the energy used is always less than the energy harvested?

2 *Maximum Performance:* While ensuring energy-neutral operation, what is the maximum performance level that can be supported in a given harvesting environment?

Several research papesr on energy neutrality tries to address these two questions in different ways, depending on performance boundaries given by application requirements [96–98]. However, there is still plenty of work to do in energy neutrality: first of all there is a need for performance optimization of motes subjected to energy neutrality constraint to increase the maximum performance level achievable. Indeed, this is one of the main limitation of energy neutrality since many real applications require high performance level in order to guarantee a boundary on delay and latency. Moreover, the integration between the energy-neutral mechanism and upper layer protocols (routing, application) it is still an open issue. This drawback need to be overcome to make energy-neutral approach an effective alternative to the lifetime maximization technique. Finally, factors that prevents the everlasting operation of motes - such as the variation of the energy availability over time and space, the limited availability of energy in the surrounding environment - have to be taken into account, as well as the compatibility of hardware components which have to guarantee the maximum energy efficiency.

As already pointed out in section 1.2, wake-up radio (WUR) technology aims to drastically increase energy efficiency of wireless sensor nodes cutting out idle listening consumption of the main radio transceiver. Since energy-neutral hardware platforms must provide the maximum energy efficiency to guarantee energy neutral operation, it seem reasonable to use radio wake-up technology in energy-neutral applications. The integration of wake-up radios in energy-neutral context is at the early stages and only few papers dedicated to the topic are available in the literature [30, 99]. For these reasons we decided to carry out a research work committed to the formal definition of the issues related to energy-neutral paradigm and to the design and implementation of novel energy neutral dedicated platforms enhanced with radio wake-up capabilities.

3.1 MagoNode++

Energy harvesting, which allows to power embedded devices from environmental energy, is currently one of the most promising technologies towards the goal of long-lasting sensing systems [100]. If the harvested energy is efficiently utilized, nodes of a WSN powered by renewable energy can achieve long-term uninterrupted operation. To reach this goal, the system energy consumption should be dynamically adapted to the availability of the energy source, so as to achieve ENO [12]. Informally, a node operates in ENO mode if the energy it consumes over a given time period is no greater than the energy harvested during the same time period. Depending on the considered scenario, achieving ENO may be challenging. For example, indoor energy sources such as ambient light in illuminated rooms can provide power only in the order of tens of μW per cm^2. To support energy-neutral operation in such scenarios, the energy consumption of the node must be reduced as much as possible. The use of a wake-up receiver (WRx) drastically increases the energy efficiency of wireless communication by virtually eliminating idle listening, which is one of the main sources of energy waste in WSNs. Motes equipped with a WRx use a dedicated ultra-low power hardware component to monitor the channel, remaining in a power-saving sleep state until other nodes in the network demand their activation.

In this section we present the MagoNode++, a novel WSN platform for energy neutral applications that combines a wake-up radio, light and thermoelectric harvesters and a battery and power management module for self-sustainable operation and long-lasting sensing systems. To the best of our knowledge, the MagoNode++ is the first WSN platform integrating light/thermoelectric harvesting, power management and radio-triggering capabilities with a power consumption of less than 3 μA with the Wake-Up Receive (WRx) in idle listening.

3.1.1 MagoNode++ Architecture

The MagoNode++ is built upon the MagoNode revision (rev.) B, a new version of the MagoNode presented in section 1.1. The MagoNode rev. B is equipped with the ATmega256RFR2 (RFR2) microcontroller and transceiver bundle, an updated version of the ATmega128RFA1 (RFA1) characterizing the previous version of the mote. The RFR2 features 256KB of ROM and 32 KB of RAM, both memories are twice the size of those of the RFA1. Furthermore, the power consumption of the RFR2 in reception (RX) is only 6 mA, half the one of the RFA1 (see section 1.1.3.1 for more details). The MagoNode rev. B features also the FT230X chip, a USB to serial UART interface which allow easy mote reprogramming and fast serial communications and an op-

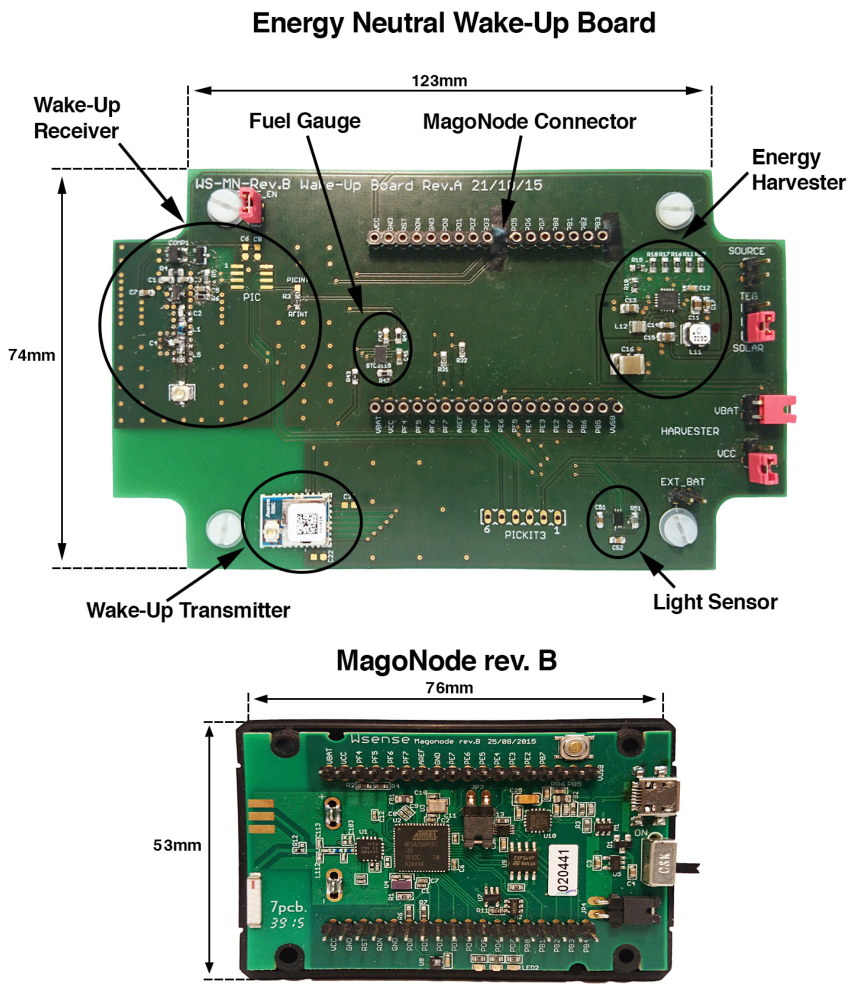


Figure 3.1: A MagoNode++ prototype with energy-harvesting and radio-triggering capabilities.

tional 2 MB flash memory. The printed circuit board (PCB) of the mote was designed to host two 18 positions 2,54 mm spacing male pin connectors where all microcontroller pads were reported (figure 3.1). This allow the sensor node to be easily attached to expansion boards, without the need to solder. For everything else, the MagoNode rev. B is equivalent to the previous version of the mote, is 100% compatible with TinyOS [52] and features the same power consumption of its predecessor.

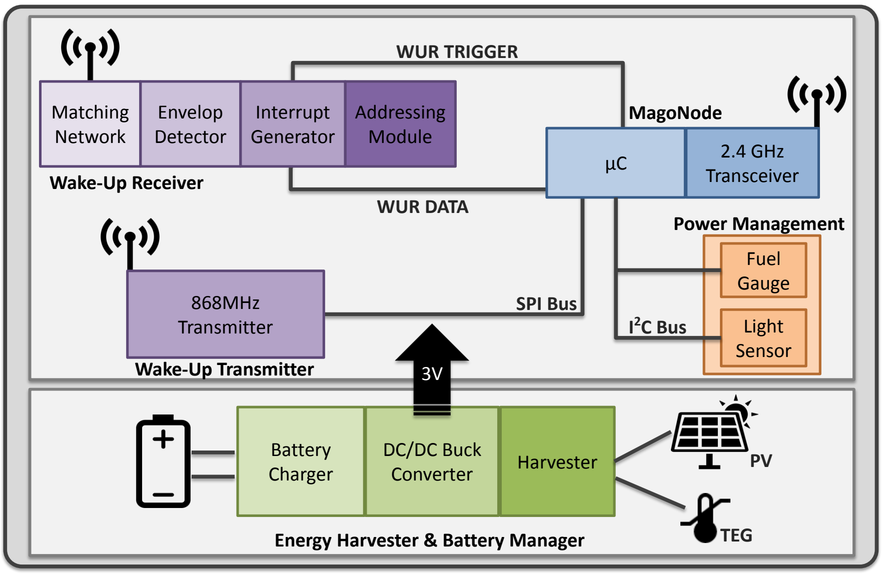


Figure 3.2: Architecture of the MagoNode++.

Energy Harvester and Battery Manager Module. The core of the energy harvesting subsystem is an energy-harvesting integrated circuit (IC) designed for low-power application, the BQ25570 by Texas Instruments. The BQ25570 efficiently converts and stores energy generated from various sources, including photovoltaic panels and thermoelectric generators, using a dynamic Maximum Power Point Tracking (MPPT) algorithm. The harvested energy is then accumulated in a storage device, such as supercapacitors or Li-Ion, Li-Polymer, NiCd or NiMH batteries. The BQ25570 is equipped with a nano-power buck converter to supply the load with a regulated voltage. Whenever the battery voltage is above a certain threshold (typically 2.8 V), the embedded buck converter is activated to supply energy to the MagoNode++.

Power Management Module. To ensure energy-neutral operation, a mote must choose next task to accomplish in such a way the energy budget, i.e. the difference between available energy and cost in term of power of the considered task, is always positive. For such reason, a energy neutral platform must feature a power management module able to provide a fine-grained estimation of the energy availability.

The most immediate way to compute the current amount of available energy is to use a fuel gauge. A fuel gauge is an IC able to estimate the battery state of charge (SOC) from voltage and current flowing inside and outside the battery. We equipped the MagoNode++ with the ST STC3115 fuel gauge which estimates the battery state of charge SOC from the battery voltage and from the current flowing from and to the battery. For solar harvesting the amount of energy available can be estimated exploiting the correlation between data coming from a light sensor and information about the amount of energy gathered from the solar panel, depending on lighting conditions. Since the MagoNode++ is equipped with a energy-harvesting IC able to gather energy from photovoltaic panels (see paragraph 3.1.1) and since solar energy harvesting provide the highest power density among harvesting systems available for WSN [24], we decided to equip the MagoNode++ with a low-power light sensor (the Rohm BH1750FVI) able to provide high-accuracy data about brightness variations. Above cited energy availability computation methods can be used in a mutually exclusive way to reduce the power consumption of the MagoNode++ or in a joint way to increase the estimate accuracy.

Wake-up Receiver and Wake-up Transmitter. The use of a wake-up radio increases the energy efficiency of wireless communications by eliminating idle listening on the main radio transceiver. The wake-up module of the MagoNode++ is composed by a ultra-low-power wake-up receiver (WRx) and by a wake-up transmitter (WTx) to send awakening requests. The wake-up system uses on-off keying (OOK) modulation and it is optimized to work at 868 MHz. The WUR module features high sensitivity (up to -50 dBm), short wake-up latency, nano-ampere current consumption (579 nA at 3 V in idle listening), and selective addressing capability (see section 1.2).

3.1.2 Experimental Results

We performed experimental measurements of the current consumption of the MagoNode++ by using a digital multimeter and supplying the system with 3 V. Results are reported in figure 3.3. The first histogram depicts the current consumption of the system in typical active states: data transmission and reception using the main transceiver, wake-up message transmission and reception, light sensor sampling and SOC estimation. The second histogram shows the current consumption of the MagoNode++ in its lowest power consumption state, denoted as Low Power Mode (LPM), and details the current drained by each individual module in this state. Results show that the MagoNode++ consumes only 2.8 μA in LPM with its WRx module in listening mode, outperforming existing prototypes such as [30] (PWmote V1 MSP430+CC2420, WRx and WTx CC1100) with a reduction of almost 50% in energy consumption. The low power consumption and energy-harvesting and radio-triggering capabilities of the MagoNode++ make it suitable to challenging application scenarios that requires high energy efficiency and self-sustainable operations.

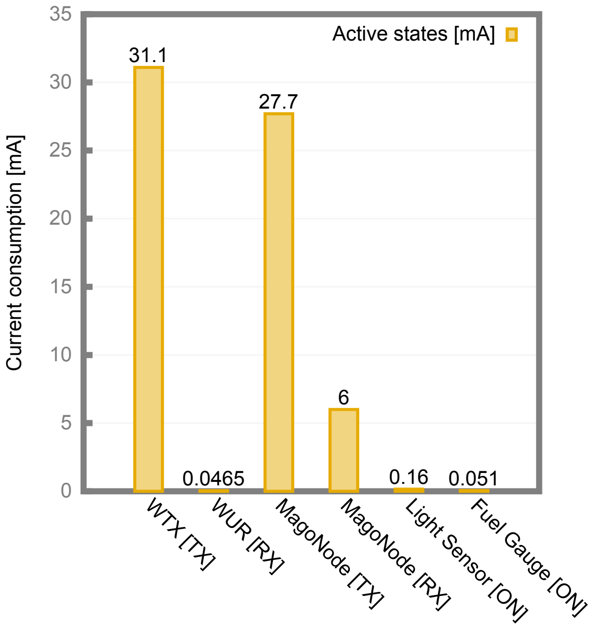


Figure 3.3: Current consumption of the components of the MagoNode++ in active states.

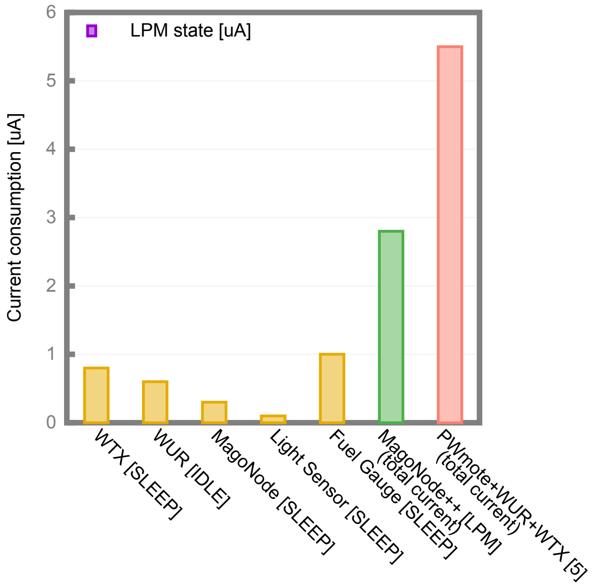


Figure 3.4: Current consumption of the components of the MagoNode++ in sleep states.

Part II

Wireless Sensor Networks For Structural Health Monitoring

Structural health monitoring (SHM) deals with the early detection of damages to civil and industrial structures, such as roads, bridges, canals, buildings and aero-space vehicles, etc., and is emerging as a fundamental tool to improve the safety and maintainability of these critical facilities. SHM requires a variety of sensing technologies together with an embedded system able to capture, log, and analyze data in order to prevent collapses and breaks, avoiding permanent damages to structures, thus simplifying and improving the effectiveness of their upkeep. Depending on application scenarios, many different types of sensors are required, including pressure sensors, vibrating-wire strain gauges, inclinometers, crackmeters, etc.

Nowadays, most of the commercial SHM systems are wired [101–103]. However, the deployment of such systems in wide areas or in harsh environments, pose both economical and practical problems. For these reasons, WSNs have been proposed as ad hoc network infrastructures to support SHM applications. SHM supported by WSN is an active and well-established research field [31–33] and some wireless SHM systems are now entering the market [104, 105].

In this part of the thesis we applied the solutions presented in part I to structural health monitoring (SHM). In particular, in chapter 4 we present the WSN we designed to monitor construction site of Rome B1 underground. The goal of the testbed was to gather structural health data during the excavation of a 700 m tunnel between two stations. Morover, in chapter 5 we introduce our work concerning the monitoring of earthquake-stricken areas exploiting WSNs. In particular, we designed and realized a network able to monitor damaged buildings and the quality of the air, which could be impaired by gas leakages. From the experience gained on-field we decided to design a new wireless, low-power, scalable hardware architecture dedicated to SHM: the Modular Monitoring System (MMS) (see chapter 6).

Chapter 4

Underground Construction Site Monitoring

The demonstration of the solutions and the technologies developed within the European project Genesi [42], concerning a wide range of hardware and software for "green" monitoring of structural health, has taken place by monitoring a construction site of the Rome B1 underground. The monitoring took place for 4 months, from the end of 2011 to the first half of 2012 using a fully battery-powered WSN. The specific goal was that of monitoring the construction of the 700 m tunnel connecting the *Conca d'Oro* and *Jonio* new metro stations. The construction involved the use of a 9 m diameter Tunnel Boring Machine (TBM) long 70 m. The excavation started at the end of November 2011 and finished in mid February 2012. The WSN used for monitoring consisted of 32 wireless nodes, 49 transducers and lasted 4 months. In the remainder of this chapter we describe requirements and solutions adopted for the deployment of that WSN and the lessons we learned from this on-field experience.

4.1 Scenario and Requirements

The monitored tunnel is divided in 4 sections. For each section, 12 vibrating wire strain gauges buried in the concrete blocks of the tunnel were used to measure deformation of the concrete to prevent collapse. The monitoring activity required 1 sample/sensor/hour, with the transmission of the sensed information to a remote server. In order to obtain data reliability, data loss could not exceed 10%, and no more than 3 consecutive packets could be lost per sensor. In addition, to add robustness, 100% of the data read from each strain sensor was expected to be recovered when physically accessing the sensor at the construction site. The monitoring system was expected to be easy to deploy and battery-powered (except for the gateway). The expected lifetime of the monitoring system should exceed one year. This constraint was introduced by SHM experts in order to increase the flexibility of the system so to adapt it to application scenarios requiring increasing network lifetime needs.

4.2 Hardware and Software

We deployed a WSN of 32 CM5000 wireless motes [106] and an Alix 3D3 embedded system with a 3G USB pen and an UPS backup system for the gateway. One of the 32 wireless motes was acting as a sink for the WSN, 8 sensing motes were interfaced to 6 strain sensors each resulting in 2 motes per monitored section. One mote was sampling temperature and humidity inside the tunnel, and the remaining 22 motes acted as relays to enable wireless connectivity. During tunnel construction the topology was continually changing since the motes were placed progressively, as the TBM advanced. The final topology at mid February 2012 is shown in Figure 4.1. Each mote was housed in an IP55 plastic box. Raytech Magic Rubber [107] was applied over the electronics for protection against the high humidity expected in the tunnel. Relay motes ran on 2x AA Alkaline batteries, providing 3 V and 1850 mAh. Due to the presence of the additional strain interface, sensing nodes were powered by 2x Type D Thionyl Chloride batteries (7.2 V, 19 Ah). Sensing nodes were placed in a steel box buried in the concrete block. Despite the Faraday effect expected from the steel box, we observed from previous experiments that the wireless signal was able to propagate few meters outside. Hence, we tried two different approaches: in the first and second monitored section we kept the antenna of the sensing node inside the steel box taking care to place a relay node nearby as signal extender. On the contrary, for the last two monitored sections, we used an external patch antenna getting rid of the additional relay node nearby, but at the expense of a lower robustness.

We used DISSense for data collection [25] on top of the TinyOS 2.1.1 distribution [52]. DISSense is an ultra-low power protocol specifically designed for periodic data collection. It takes advantages of high sampling periods for energy efficiency, and fits well in our scenario, where data are collected from each sensor once an hour. For this reason, DISSense can feature a very low duty cycle, enabling the WSN to keep running on batteries for years. For debug and statistical purposes, half of the relay nodes also generated packets at the same rate, carrying debug information (Figure 4.1). DISSense has been extended with additional features for supporting network reboot and local logging.

Each mote stored data on a flash EEPROM, which could be downloaded wirelessly by getting close to the mote (single-hop connection). Finally, the

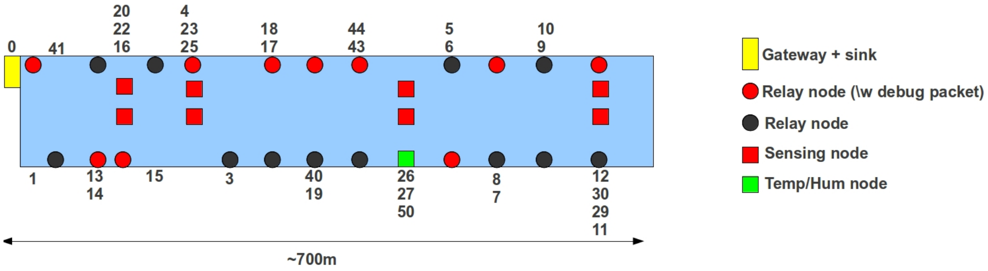


Figure 4.1: Final WSN topology (February 21 2012).

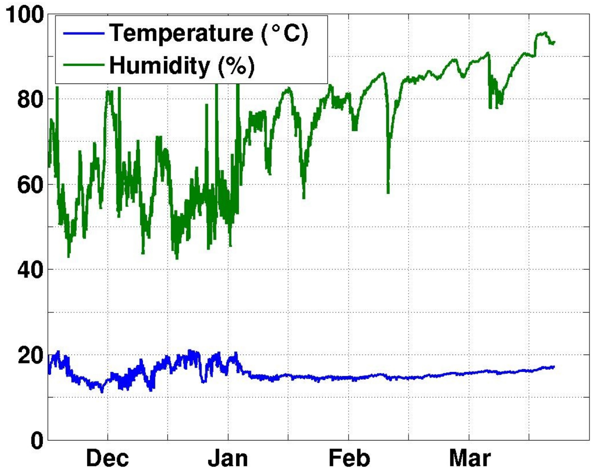


Figure 4.2: Node 50: Temperature and humidity values (December 13 2011 to March 27 2012).

gateway was equipped with auto-starting scripts enabling autonomous recover from power cycles and periodical transfer of the collected data to a remote server.

4.3 Testbed Results

We collected data on radio duty-cycle and the Data Delivery Ratio (DDR), namely, the percentage of data packets transmitted by a mote that actually is received by the sink. Data was collected on the topology in Figure 4.1 from February 22 to March 21 2012. As per the DISSense implementation the radio duty cycle is the same for all the nodes. The observed average duty cycle has been 0.22%, which confirms the effectiveness of such a low-power communication approach for the application requirements. Based on DISSense design, we were expecting high values of DDR for nodes closer to the sink, with a gradual decrease in DDR proportional to the hop distance. Surprisingly, while the highest DDR matched our expectations, with 99.9% DDR for node 1, the lowest values have been measured in the middle of the tunnel. In particular node 40 and 44 showed a DDR of 90.8% and 92.5%, respectively. Getting farther in the tunnel, the DDR increased again to 99.1% for node 30. This is because the "quasi linear" topology of our testbed induces a high number of collisions in the middle of the tunnel, due to the hidden terminal problem. Despite data loss, the values from all nodes and from the temperature and humidity sensor (node 50) were retrieved at the end of the testbed experiments on March 27 2012. The actual values cannot be published due to the constrained imposed by an Non-Disclosure Agreement (NDA) with Roma Metropolitane S.p.A. Therefore, in Figure 4.2 we plot the temperature and humidity values sampled by node 50. It is interesting to note the high volatility of the values especially from December to mid January. This is because node 50 was initially installed on the TBM and, consequently, the sampled values were affected by the activity of the machine. From mid January, when the TBM had almost finished the excavation, we moved the node to its final place, shown in Figure 4.1.

4.4 Concluding Remarks

The technical solutions we adopted for the testbed were successful, in that everything worked and data conveyed by the network were consistent with those offloaded by the sensors at the end of the testbed experiments. The lessons learned include the following. The UPS and auto-starting scripts of the gateway are critical: we experienced several power failures and, in three circumstances, the power cut was long enough to make the UPS shutdown. In all cases, the system autonomously recovered as soon as the power came back because of the auto-starting scripts. The remote reboot procedure added to DISSense has been extremely useful to recover from a bug that made the network to partition. Finally, as confirmed by the humidity levels shown in Figure 4.2, protection for electronic is mandatory and the MagicRubber we applied to all the nodes has successfully proven the right tool for the job.

4.5 Acknowledgments

We thank Roma Metropolitane and Treesse Engineering [108] for the opportunity to access the construction site. The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement Infso-ict n.257916, Genesi project [42].

Chapter 5

Earthquake-stricken Areas Monitoring

In this chapter we introduce our work concerning SHM in earthquake-stricken areas. In particular, we designed and realized a WSN featuring sensors able to monitor damaged buildings and to acquire information about the quality of the air, which could be impaired due to gas leakages. All the work done was carried out in the context of the research project “TETRis - TETRA Innovative Open Source Services” [34]. The main goal of TETRis was to extend the public safety services and applications running on top of the Terrestrial Trunked Radio (TETRA) [35] communication system with real-time data collected by WSNs deployed on field. The reference architecture for TETRis is shown in figure 5.1. At the lower layer, the *Networks* - and TETRA in particular - are used to support the communication among the *Smart Objects*. A smart object, is a physical/digital object, augmented with processing and networking capabilities. The *Smart Objects* layer coordinates and orchestrates the interaction among the *Smart Objects* to support the provisioning of value added services to the *Smart Environments*, namely “a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network” [109]. In this architecture, each WSN - and in particular the collection point (i.e. sink node) of the sensor network - is interfaced with a smart object and provides real-time data employed to support the provisioning of advanced services for the Smart Environments.

5.1 Application Scenario and Requirements

The reference application scenario envisioned for the TETRis project is the following: a rescue team, while exploring an earthquake-stricken area, deploys

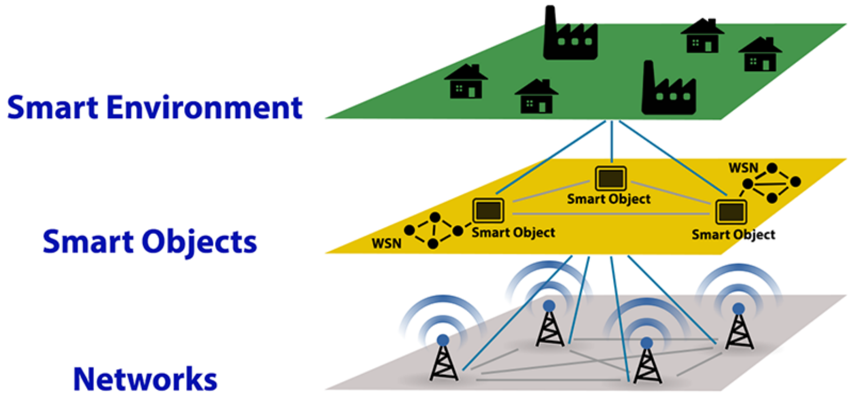


Figure 5.1: A reference architecture for TETRis networks and services.

one or more WSNs capable to collect real-time data that can prevent or limit future injuries or damages to the population and to the infrastructures. Deployed WSNs send gathered information to a smart object, which, in its turn, forwards data to a control and management point using the TETRA network. In this specific application scenario the use of TETRA as relay network is crucial since mobile network could be not operational because of the earthquake. Furthermore, another important application requirement concerns the reliability of radio links and the ability of covering large areas with a limited number of nodes. These features are vital for the effective adoption of WSNs for long-lasting monitoring activities in areas affected by natural disasters. Indeed, while in the literature WSNs are usually assumed to be densely deployed, in this specific application scenario and, more in general, in real deployments, the size of the network is limited by practical considerations such as: cost constraints, limited number of sensing points necessary to support effective monitoring and limited accessibility to disaster areas.

5.2 TETRis WSN Architecture

We decided to use the MagoNode (see section 1.1) as sensor node of the WSN we realized in the context of TETRis project. We made this choice because the MagoNode features can be easily exploited to fulfill TETRis application scenario constraints. Indeed, the MagoNode is a state-of-art low-power platform which, being compatible with TinyOS, supports a number of energy-aware MAC and routing protocols. Exploiting those features, a MagoNode supplied with two AA type 2000 mAh batteries could last for a period of time between months to years, this is compliant with TETRis application scenario.

During the design of the WSN in the scope of TETRis project, we decided to exploit two well-known energy efficient protocols available in TinyOS 2.1.2 release: the Box-MAC protocol featuring Low Power Listening (LPL) [55] and the Collection Tree Protocol (CTP) [56]. The Box-MAC is a duty-cycle MAC protocol, that increases the energy efficiency of the communications lowering the idle listening time of receivers, that as already observed, is a major component of energy expenditure in wireless communications. CTP is a reliable and robust routing protocol for the collection of the data sensed by motes. CTP is a tree-based collection protocol: nodes in the network form a set of routing trees in order to reach nodes that advertise themselves as tree roots. The selection of the most "reliable" route is made using the expected transmissions metric (ETX), used as routing gradient. We selected LPL and CTP because they have been widely tested in various projects and represent a de facto standard for data collection in long-lasting WSNs. Furthermore, the MagoNode is equipped with a RF front-end which enhances wireless performance, extending mote radio range and increasing wireless links reliability. This is a crucial feature for the effective monitoring of areas affected by natural disasters, where the deployment of dense WSNs could not be feasible. Finally, the MagoNode, featuring a modular architecture, can be easily integrated into expansion boards featuring application specific sensors and transducers necessary to support the TETRis application scenario (see subsection 5.2.1 for details).

5.2.1 Sensors for TETRis Application Scenario

We designed three MagoNode expansion boards featuring additional sensors and transducer necessary to support the TETRis application scenario. In the specific, we realize a board to track the quality of the air, a board to monitor critical infrastructures and one to collect data about temperature and humidity.

MNA Ambient Board for Air Quality Monitoring. The MNA Ambient Board (figure 5.2) is used to monitor the quality of the air and it is equipped with four sensors: Co2Meter COZIR CO2 sensor, Sensirion SHT75 temperature and humidity sensor, Sharp GP2Y1010AU0F dust sensor, Figaro TGS2442 CO sensor. All those sensors are characterized by limited power consumption, making them appropriate for the long-lasting deployments envisioned in the TETRis scenario. The COZIR is an ultra low-power (3.5 mW) digital CO2 sensor with a measurement range between 0 ppm and 2000 ppm. It features a low noise measurement (<10 ppm), low peak current (33 mA) and fast warm-up time (<3 s). The Sensirion SHT75 is a digital temperature and relative humidity sensor. It provides a fully calibrated digital output with a precision ranging from 8 to 12 bits and is characterized by a 0.5mA current consumption during measurements and only 0.3 $/mu$A in sleep mode. The Sharp GP2Y1010AU0F is an analog dust sensor which detects the reflected light of dust in air by means of an infrared emitting diode and a photo-transistor with a limited current consumption of 11 mA. The Figaro TGS2442 is a low-power, CO analog sensor with a measurement range between 30 ppm and 1000 ppm. It features low sensitivity to alcohol vapor and high sensitivity to carbon monoxide.

MNA MultiSensor Board for Structural Health Monitoring. The MNA MultiSensor Board (figure 5.2) is meant to interface a great variety of analog sensors including those dedicated to structural health monitoring, like crack-meters, strain gauges and inclinometers. In our testbed we used the D313 crack-meters and the VK40 vibration wire strain gauge by SISGEO [110]. In order to acquire data from analog sensors the MNA MultiSensor Board is equipped with a high precision, high data rate Analog to Digital Convert (ADC) by Texas Instruments, the ADS1256 and some additional instrumentation amplifiers, comparators and precision voltage references. The ADS1256 is a low-noise, 24-bit ADC converter with a maximum output data rate of 30 kSPS and eight single-ended inputs or four differential inputs.

MNA Temp&Hum. We also implemented the MNA Temp&Hum Board a simple temperature and humidity sensor board developed mainly for prototyping and debugging purposes and to serve as relay node in the network.

5.3 TETRis Testbed

In this section we discuss the results of a deployment in an hybrid indoor/outdoor testbed that is consistent with the reference TETRis scenario, presented in section 5.1. This testbed was used as one of the final demonstrators of the TETRis project. An interview with engineers experts on structural health monitoring provided us “minimalist” requirements for the considered scenario: only few sensors per buildings are necessary and a sample per node every 15 minutes is considered sufficient to provide meaningful measurements.

The testbed is made by 11 nodes placed at the basement and at the roof of the Department of Computer, Automatic and Management Engineering (DIAG) of Sapienza - University of Rome. All the protocols (i.e., LPL and CTP) were tuned to optimize the energy consumption, and consequently the network lifetime, in view of the sampling requirement.

Indoor nodes are two MNA Ambient Boards, featuring CO, CO2 and dust sensors, and thus capable to collect a number of relevant data on the quality

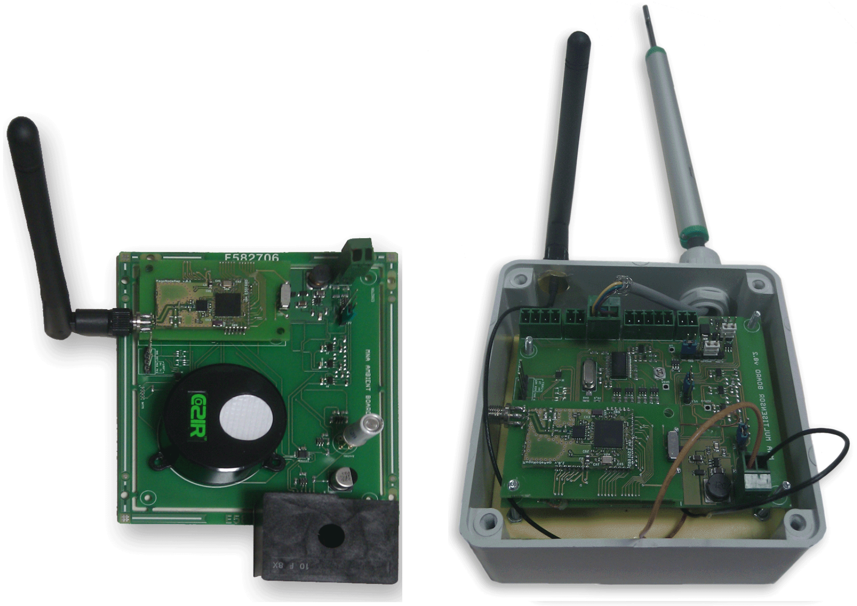


Figure 5.2: The MNA Ambient Board (left) and the MNA MultiSensor Board inside a IP56 box (right)

of the air in two adjacent rooms. Four MNA Temp&Hum Boards (see table 5.1) sampling temperature and humidity, are placed along the corridors. The structural health monitoring activity is performed by two MNA MultiSensor Boards: one is placed indoor and it is interfaced to a vibrating wire strain gauge, while the other is placed outdoor and it is connected to a displacement sensor. Finally, two additional MNA Temp&Hum Boards acting as relay nodes were deployed outdoor to manage wireless connectivity between the basement and the roof.

The LPL wake-up period of each node is set to 500 ms and the listen time is set to 5 ms. This allows us to achieve a base duty cycle of the radio of 1%, which is sufficient to guarantee a network lifetime in the order of months. Notice that, the overall duty cycle is slightly larger than 1% (see table 5.3) due to events that cannot be fully controlled, such as the traffic generate by the CTP routing layer and the retransmissions of data packets. The overall power consumption budget has also to take into account the heterogeneous consumption of the hardware interfaces described in section 5.2.1. In particular, the CO and CO2 sensors are by far the most energy demanding. However, due to the relatively large sampling interval suggested by the domain experts, the energy necessary to sensing is marginal with respect to the energy employed in wireless communications. Finally, we enclosed the outdoor nodes into IP56 boxes, in order to protect the hardware from the agents of weathering. The impermeability of the casing is mandatory since the network will be deployed in harsh environmental condition. Table 5.2 summarizes all the settings used in the testbed.

The testbed started on December 2013 and ran 1 month. In this period, outdoor sensors have been exposed to 8 days of rain with temperatures ranging between 2 to 15 Celsius degrees. The results of the testbed, shown in table 5.3, point out the high reliability of the Collection Tree Protocol that guaranteed a Data Delivery Ratio above 99.5% for all the nodes of the network. High reliability is a crucial requirement in public safety scenario in order to elaborate trustworthy statistics on the state of the monitored area. As expected, the observed duty cycle ranges between 1.13% and 1.24% and it is thus slightly larger then the base one (1%). However, such increase does not impact significantly on the expected network life-time that is in the order of months.

Finally, the average network diameter, i.e., the average number of hops that a packet has to traverse to reach the sink, is only 2.2. This is a remarkable result, considering that the radio signal has to traverse concrete walls with thickness ranging from 70 cm to 100 cm, and a number of metal structures. Furthermore, the number of parent changes in the CTP protocol is significantly low; nodes change their parent due to bad links only in few occasions. This is an indirect evidence of a good network stability. The combination of small network diameter and good link stability is a direct consequence of the improved link quality provided by the RF front-end of the MagoNode platform.

Table 5.1: TETRis testbed hardware composition.

Id	Board	Sensors	Power	Note
0	MNA Board	N/A	USB 5 V	Sink, Indoor
2	Ambient Board	CO,CO2,Temp.,Hum.	4x AA Alk.	Indoor
3	MNA Board	Temp., Hum.	4x AA Alk.	Indoor
4	MNA Board	N/A	2x AA Alk.	Relay, Outdoor
5	MNA Board	Temp., Hum.	2x AA Alk.	Indoor
6	MNA Board	Temp., Hum.	2x AA Alk.	Indoor
7	MNA Board	Temp., Hum.	2x AA Alk.	Indoor
9	MNA Board	N/A	2x AA Alk.	Relay, Outdoor
11	Multisensor Board	VW Strain Gauge	4x AA Alk.	Indoor
15	Multisensor Board	Displacement	4x AA Alk.	Outdoor
19	Ambient Board	Dust,CO2,Temp.,Hum.	4x AA Alk.	Indoor

Table 5.2: TETRis testbed parameters.

Parameters	Value	Parameters	Value
Inter-Packet Interval	900 s	LPL Wake-up Period	500 ms
Packet Size	50 bytes	LPL Listen	5 ms
Run Time	15 days	LPL Delay After Receive	20 ms
Nodes Number	11	RF Channel	16

Table 5.3: TETRis testbed results.

Id	DDR (%)	Duty Cycle (%)	Hop Count	Parent Changes
0	100	100	0	0
2	100	1.13	1	0
3	100	1.21	1.93	4
4	99.65	1.22	2.73	3
5	100	1.2	1.68	4
6	100	1.2	2.02	4
7	100	1.17	2.06	4
9	99.86	1.19	3.73	1
11	100	1.18	1	0
15	99.86	1.21	4.58	6
19	100	1.24	1	0

5.4 Concluding Remarks

In this section we described the work done in the scope of the TETRis project, to design a WSN aimed to extend the functionalities of TETRA, in order to better support the management of critical situation with real-time feedback acquired on the field. In the specific, in section 5.2 we introduced a WSN able to monitor damaged buildings and gas leakages in earthquake-stricken areas based on the MagoNode mote (see section 1.1). The MagoNode features low-power consumption and outstanding RF performance thanks to the embedded RF front-end which enhances radio capabilities in both transmission and reception. Moreover, the MagoNode, thanks to its modular architecture, can be easily integrated into expansion boards featuring sensors and transducers necessary to support the TETRis application requirements. Finally, the MagoNode is compatible with TinyOS 2.1.2 which supports various well-known energy efficient communication protocols.

In section 5.3 we described the deployment of the above cited WSN in a hybrid indoor/outdoor testbed consistent with the reference TETRis scenario. The testbed started on December 2013 and ran 1 month. The results of the testbed point out the high reliability of the Collection Tree Protocol which guaranteed a Data Delivery Ratio above 99.5% for all the nodes. High reliability is a crucial requirement in public safety scenario in order to elaborate trustworthy statistics on the state of the monitored area. Furthermore, the average network diameter, i.e., the average number of hops that a packet has to traverse to reach the sink, is only 2.2. This is a remarkable result, considering that the radio signal has to traverse concrete walls with thickness ranging from 70 cm to 100 cm, and a number of metal structures. Finally, the number of parent changes in the CTP protocol is significantly low. This is an indirect evidence of a good network stability. The combination of small network diameter and good link stability is a direct consequence of the improved link quality provided by the RF front-end of the MagoNode platform.

Chapter 6

Modular Monitoring System

Nowadays, most SHM systems in the market are wired. However, the deployment of a wired system in a wide area or in a harsh environment, can pose both economical and practical limitations. For this reasons, WSNs have been proposed as an ad hoc network infrastructure to support SHM, avoiding prohibitive costs of wired systems and easing the on-field deployment. SHM supported by WSN is an active and well-established research field and some wireless SHM systems are now entering the market. The work presented in this chapter introduces the MMS, a wireless, low-power, scalable hardware architecture dedicated to SHM. A key feature of the MMS is its high modularity that allows easy customization of the platform depending on the number and the type of sensors required by the specific application scenario. In addition, MMS fully supports multi-hop wireless communication paradigm and mesh networks.

The remainder of this chapter is organized as follows. Section 6.1 presents the state-of-the-art on both wired and wireless SHM systems. In section 6.2 we explain the motivations supporting the design choices made during the development of the MMS, while in section 6.3 we introduce the features characterizing our system. Furthermore, in sections 6.4 and 6.5 we present the hardware prototype and the achieved results respectively. Finally, in Section 6.6, we give conclusions and directions for the future.

6.1 State of the Art

As already pointed out, most SHM systems available on the market, such as the Geomonitor by Solexperts [101], are wired. However, deploying those systems can be cumbersome: besides the installation costs, a detailed deployment plan is required to face evolving needs of different construction phases. Moreover, cables are subjected to accidental cuts and damages and, in some scenarios, their installation is unfeasible or inappropriate (e.g., historical buildings). Along with wired systems, some standalone data loggers dedicated to SHM are commercially available, among the others: Solexperts SDL [101], Geokon 8002-16-1 [102] and Keller GSM-2 [103]. These systems are simpler to install, but they do not allow real-time remote monitoring and require frequent in situ access by qualified personnel to collect data. The introduction of wireless communications in SHM gives immediate advantages in terms of easier deployments and reduced maintenance and personnel costs. However, when monitoring devices are battery-powered, the use of wireless communications is among the most energy demanding feature that can significantly limit the devices lifetime. Prominent examples of wireless monitoring systems are: National Instruments Wireless Data Acquisition (WiFi-DAQ) [104] and MicroStrain wireless sensor network [105]. While the former supports IEEE 802.11 standard, the latter is compliant with IEEE 802.15.4. Both products adopt a conservative approach by limiting wireless communication to 1-hop. Supporting multi-hop wireless communications was investigated in several research papers in the last decade [31] [32] [33]. Multi-hop networking provides a number of advantages: scalability (larger areas can be covered), reliability (failure and multiple routing paths without single point of failure) and ease of deployments (the presence of multi-hop relay nodes allows to bypass obstacles like walls and metal structures). Recently, National Instruments presented the NI WSN [111]: a multi-hop battery-powered WSN supporting up to 36 nodes configured in a mesh network and up to 3 years node lifetime.

6.2 Motivation

As seen in the previous section, WSNs are slowly entering the market of SHM applications. An attempt to develop a robust solution and to test it in realistic application scenarios was made in the Genesi Project [42]. The main goal of this project was to design and implement a “novel generation of green wireless sensor networks which can be embedded in buildings and infrastructures at the time of construction and be able to provide a monitoring and control intelligence over the whole structure lifetime”. The project involved the monitoring of a bridge construction site in Fribourg, Switzerland [112] and the construction of a tunnel for the B1 underground line in Rome, Italy (section 4), by means of WSNs. The outcomes of these experimental activities, highlighted the advantages of WSN technology compared to traditional monitoring techniques. According to the application requirements provided by the SHM experts, Genesi nodes can support a number of heterogeneous sensors. However, a Genesi node can manage only a single sensor of the same type, while there are some applications in which multiple instances of the same type of sensor are needed. As an example, in a 3-axis deformation analysis, a single wireless node may need to interface with three instances of a vibrating-wire strain gauge while to monitor a concrete junction the node may need to interface a current-loop inclinometer sensor and a resistive displacement sensor. Other commercial solutions, such as the NI WSN described in section 6.1, can handle multiple instances of the same sensor but can not support different sensor families at the same time. In general, the development of new ad hoc devices addressing the specific requirements of an SHM deployment is impractical, while the adoption of commercial solutions in such contexts is not optimal both in terms of flexibility, size and costs. This brought us to propose a low-power slotted modular system made by a set of modules connected through an internal communication bus. This solution features one wireless master module managing a group of extension modules, each one designed to interface a specific sensor. The flexibility of the proposed architecture, named MMS, allows us to support a vast number of SHM applications by simply plugging into each node the required extension modules. By changing the master module, it is also possible to easily modify the wireless technology, thus effectively adapting to the heterogeneous needs of indoor and outdoor communication requirements.

6.3 System Architecture

The original MMS architecture presented in [113] was based on a master/slave communication abstraction where "a *master module*, provided with radio capability and responsible for most of computational tasks, communicates through a low-power shared bus with a maximum of four *extension modules* (slaves)". The new architecture embraces the same principle but required the design of a new low-power shared bus.

In particular, the low-power shared bus in [113] was based on the Serial Peripheral Interface (SPI) Protocol with dedicated Chip Select (CS) and interrupt lines for each module. The rationale for this choice was to minimize the power consumption of the whole platform by allowing the master module to selectively activate each extension module acting on the corresponding chip select line, thus, without affecting the power consumption of the other modules installed on the platform. Despite the effectiveness of such solution from the power consumption perspective, the elevated number of dedicated lines (2 shared lines for the power supply, 3 shared lines for the SPI and 2 dedicated lines, CS and interrupt, for each extension module) forced us to design a backplane board with a fixed number of slots supporting one master module and 4 extension modules (figure 6.1).

This choice quickly revealed its limits: the additional backplane board

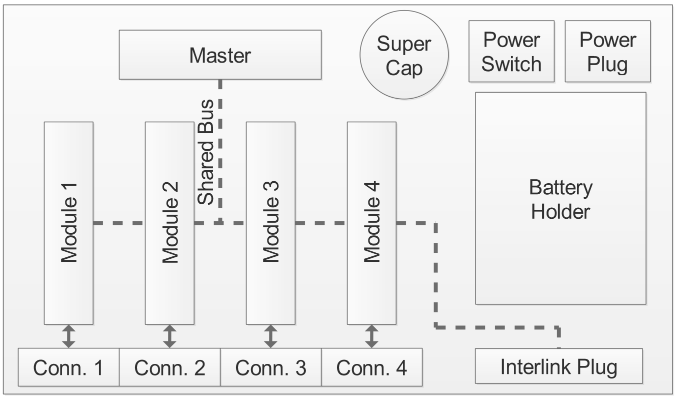


Figure 6.1: First version of the Modular Architecture

increased the overall size of the platform and its costs for the electronics, and it also required extra costs for the ad hoc housing. To overcome these limits we took hints from existing modular housing solutions available on the market and we updated our design as described below. We found out that most of the considered modular products are based on O type DIN rail standard (DIN EN 60715), a widely used solution for mounting circuit breakers and industrial control equipment inside racks. To ease the communication between modules, such solutions commonly provide a 5 lines shared bus. Hence, we implemented our system in a modular DIN housing and we redesigned the low-power shared bus to be compliant with the reduced number of lines offered by DIN bus. This led us to the new MMS architecture described in the following paragraphs.

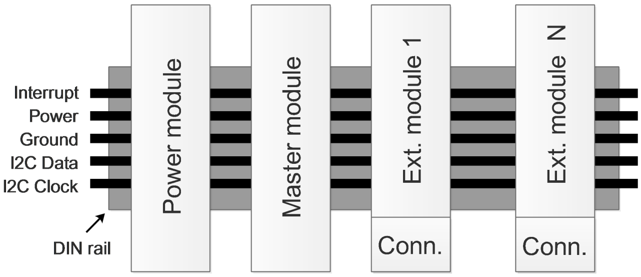


Figure 6.2: New Modular Architecture

Low-Power Shared Bus. The availability of only 5 shared lines in the bus drastically reduces the implementation options for a master/multi-slave communication system. Excluding power and ground lines, the low-power shared bus has to be implemented on the remaining 3 shared lines. To overcome this constraint, we used the Inter Integrated Circuit protocol (I^2C) which is a multi-master/multi-slave serial communication protocol based on two open-drain lines bus [114]. The communication protocol, which is master initiated, is based on the transmission of a 7 bit address, i.e., the slave address to which the master wants to communicate, followed by a read/write bit and a set of protocol dependent commands. The third open-drain shared line has been used by the extension modules to trigger the master to start a communication. The master module polls to slaves to identify which extension module triggered the interrupt on the third line. Figure 6.2 shows the new architecture with a detailed view of the new low-power shared bus implementation.

This new solution easily scales with the number of extension modules (the standard implementation of the I^2C bus is designed to support up to 127 devices), thus increasing the flexibility and optimizing the size of the MMS. However, from the power consumption perspective, we need to pay attention to several drawbacks:

- **Wake-up.** When the master module issues an I^2C start command, all the extension modules wake up to perform the address matching, even if the communication is addressed to one module only.
- **Polling.** When an extension module starts a communication, it notifies the master using the interrupt line. Subsequently the master polls all the devices to find the initiator.
- **Latency.** The I^2C protocol needs more data to be sent over the bus with respect to the SPI one, this increases the latency and the power consumption of the MMS.

In subsection 6.5.3 we analyzed each of these aspects measuring the power consumption overhead related to this new implementation with respect to the original low-power shared bus. The results show that the overall consumption of the MMS is marginally affected by this choice.

Master Module. As in the first version of the MMS, the master module is responsible for managing the wireless connectivity within the WSN and manages the extension modules. When switched on, the master performs a discovery routine for dynamically assign the I^2C addresses to each extension modules and to retrieve the configuration information from them. Then, the master computes the sensing schedule and switches to the operational mode. Below a typical master-initiated interaction with the extension module is described:

1. The master issues an I^2C start command over the bus and sends the address of the extension module followed by a write bit and a data request command.

2. The extension module wakes-up and starts the conversion while the master enters a low-power state, waiting for an interrupt.
3. The master wakes up when an extension module pulls down the interrupt line.
4. The master polls all the modules searching for the initiator.
5. When found, the master sends a *Data Read* command to retrieve the new data.
6. When the read finishes the extension module puts the interrupt line in high impedance and enters sleep state again.
7. The master module checks the interrupt line level for other extension module that are willing to communicate. If the line is still low, it jumps to step 4 otherwise it goes back to sleep.

Extension Module. The extension modules provide the hardware interface to external sensors. Each module communicates over the low-power shared bus with the master by means of an I^2C-capable microcontroller. When powered-on, the extension modules participate to the I^2C address assignment managed by the master module. Once an extension module gets an address, it switches to a sleep state and wakes up only upon the detection of an I^2C start command on the bus. The extension module can initiate a communication by pulling down the shared interrupt line of the low-power shared bus, which activates the polling procedure by the master.

Each extension module provides a register configuration area that enables the interaction with the master module. The register area is divided in 5 subsections as follows:

- The **Information area** [**read-only**] stores information such as device type and revision. This area allows the master to identify the extension module, e.g., sensors supported, channels available, etc.
- The **Command area** [**write-only**] is used by the master to trigger commands to the extension module such as a read command.
- The **Settings area** [**read/write**] stores settings for each sensor channel such as the sensor type connected, the periodic sampling value, etc.
- The **Channel flags area** [**read-only**] is used by the extension module to notify that a new data is available.
- The **Channels data area** [**read-only**] is used by the extension module to store the last sensor value read.

An example of extension module register implemented on our test board is shown in table 6.1.

The hardware interface provided by each module depends on the sensors it supports: it can be fully digital, e.g., to interface RS-232 or RS-485 peripherals, or analog, to connect sensors such as current-loop, vibrating wire strain gauges, resistive etc. Each extension module can support different kinds of sensors or several sensors of the same type.

Table 6.1: Extension module register area example.

Address	**Register**
0x00	Device type
0x02	Device description
0x03	Cmd channel 0
0x04	Cmd channel 1
0x05	Cmd channel 2
0x06	Cmd channel 3
0x07	Configuration channel 0
0x0e	Configuration channel 1
0x17	Configuration channel 2
0x1f	Configuration channel 3
0x27	Data flag channel 0
0x28	Data flag channel 1
0x29	Data flag channel 2
0x2a	Data flag channel 3
0x2b	Data channel 0
0x3b	Data channel 1
0x4b	Data channel 2
0x5b	Data channel 3

Power Module. The modular design of the new MMS allows us to support a number of interchangeable power modules which provide the power supply to the whole MMS through the power line on the bus. The modules support any kind of battery type providing a voltage between 3 and 5.5.5 V as well as the 240 V AC and energy harvesting solutions. Notice that this flexibility was not possible in the previous release of the MMS, since the power unit had to be integrated into the backplane at design time. Moreover, this feature enables the system to integrate alternate power sources such as 240 V AC main transformers or energy harvesting solutions, without the need to provide them externally as needed in the first version.

6.4 Hardware and Firmware Development

We developed and assembled the MMS at the Department of Computer, Control and Management Engineering of the University of Rome “La Sapienza”. This new version includes one master module, two demo extension modules and two power modules. As opposed to the first prototype presented in [113], this version fits in a complete modular housing solution based on pluggable slots and snap-in connections for external sensors (figure 6.3). The whole system was housed in a steel IP67 enclosure with a mounted DIN rail, an external antenna and IP68 cable glands.

Master module. The master module is responsible for radio communication and for managing the extension modules. It is based on the MagoNode (see section 1.1, a wireless hardware platform specifically designed for WSN applications. The master module is equipped with a NOR flash for persistent data storage, three status leds, a mini-usb plug and an RP-SMA (Reverse Polarity SubMiniature version A) antenna connector. An optional wireless HART [115] communication module can be installed based on application requirements.

The firmware is written using TinyOS [52], an event-driven, open-source operating system (OS) dedicated to WSNs. TinyOS is designed to cope with typical constraints imposed by WSNs: low computational capabilities, limited memory and scarce energy resources. TinyOS also provides a set of libraries implementing low-power wireless protocols for medium access control [28] and routing [56]. Hence, the ability for the master model to support the TinyOS operating system enables a straightforward support for the low-power WSNs communication abstraction.

Demo Extension Module. We developed a general-purpose extension module for demo applications which provides 4 24-bit analog channels, 2 excitation current outputs and 2 external voltage references. The device logic is handled by an efficient ARM Cortex M0+ micro-controller (MCU). The MCU provides the I^2C hardware interface and an interrupt channel required by the low-power shared bus communication, an SPI interface for the 24 bit ADC and a set of General Purpose Input/Output (GPIO) pins for driving the 3 status leds of the module. As opposed to the first MMS prototype which could switch off the extension modules, in this case the low-power shared bus only provides an always-on power line. This forces all the extension modules to remain (by default) in a sleep mode when inactive: they are enabled only when an address match event on the I^2C peripheral occurs. The demo extension module supports a sleep mode with a current consumption of 1.5 μA[1]. As the master module, the firmware was implemented using the TinyOS operating system which provides the primitives for I^2C and SPI communication and for handling the sleep mode of the device.

Power Module. We currently support two basic power modules: an AA and D type 3.6 V thionyl chloride battery. A module to power the MMS via the 240 V AC is in an advanced development stage and we are also working on the integration of an energy harvesting module supporting solar cells.

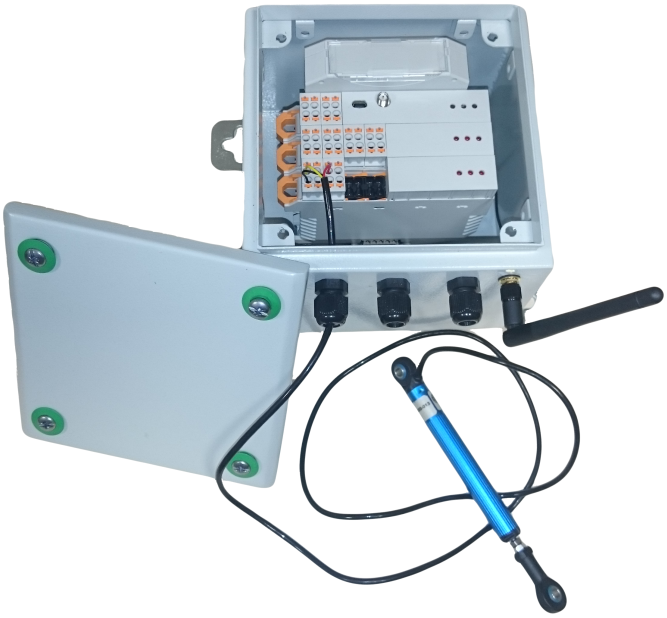


Figure 6.3: The MMS with a power module, a master module and two extension modules.

6.5 Experiments

In our experiments of the new MMS design, we first measured the power consumption during a local data acquisition (i.e., with wireless radio disabled), then we evaluated the system in a WSN testbed. The former measurement validates the effectiveness of the low-power design in our architecture, while the latter demonstrates how the MMS features energy consumption levels suited

[1] An errata specific to the MCU revision used in the extension module avoided to reach this value which refers to the newest chip revisions.

for WSN applications. Finally, we performed a set of measurements aimed to demonstrate how the energy consumption overhead introduced by the new design of the MMS (see section 6.3) is negligible if compared to the overall system consumption.

6.5.1 Local Data Measurement

All the measurements were done powering the MMS system with a Rigol DP1308A programmable DC power supply, providing $3V$. The MMS was connected in series to a Rigol DM3068 digital multimeter that sampled the current consumption at $10kHz$. The measurements were taken from an MMS made of a master module and a single demo extension module connected to a displacement sensor, i.e., a potentiometer with $4k\Omega$ series resistance. We performed a single master-initiated sampling request which consists in the following steps:

1. The master module sends the address of the extension module and sets a bit in the command area register to trigger one sample from a single channel on the selected extension module.

2. The extension module starts the conversion while the master goes back to sleep waiting for an interrupt.

3. The extension module generates an interrupt to the master notifying that it has a new data sample available.

4. The master polls all the modules, reading the data flag area. Whenever a flag notifies new data available, the master retrieves the new data.

In a second experiment, we repeated the same procedure using an additional demo extension module connected to an identical transducer where the master performs a sample request from each module. The current consumption of both tests are shown in figure 6.4. The data conversion time, which is equal to 50 ms, is determined by the sampling rate of the ADC which was configured to 20 samples per second. As expected, the current consumption during data conversion is doubled in the two extension modules configuration since each module performs the conversion at almost the same time. In addition, it is clearly visible how the sample request and data retrieval phases last twice in the two extension modules configuration since the master module needs to perform these activities sequentially for both modules. In both MMS setups, the system automatically switches back to sleep mode as soon as the data retrieval phase is completed.

Table 6.2 summarizes the average current and energy consumption of each phase measured on both configurations. Note that for the two extension modules configuration there are overlapping effects between phases: when the master module sends the start conversion command to the second extension module, the first module is already converting. Similarly, during the data retrieval phase, when the master module starts to download the new data from the first module, the second one is still converting. This explains the differences in the average current consumption during the sample request and data retrieval phases of the two modules configuration with respect to the single one.

Table 6.2: Local measurement - consumption details.

	Single ext. module		Double ext. modules	
State	Avg current (mA)	Energy (μJ)	Avg current (mA)	Energy (μJ)
Sleep	0.0035	N/A	0.005	N/A
Sample request	3.65	28	5.23	67
Data conversion	1.69	281	3.36	543
Data retrieval	2.79	104	3.57	178
Total	1.95	413	3.51	788

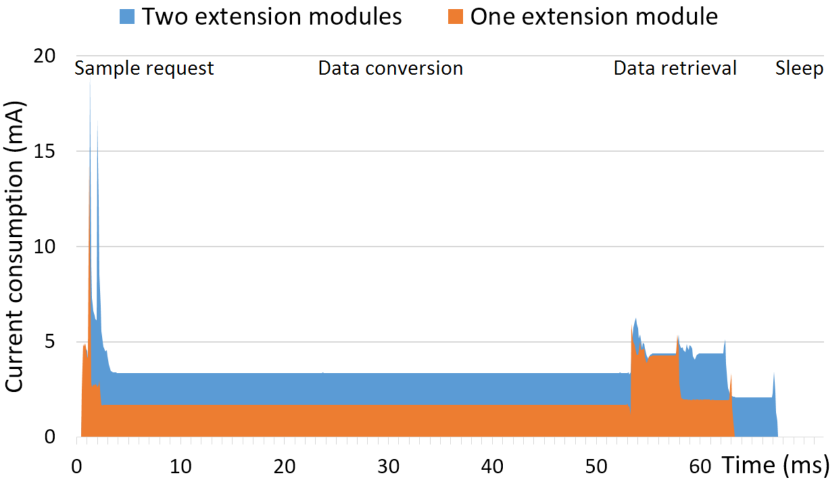


Figure 6.4. MMS current consumption during local data acquisition.

6.5.2 WSN Testbed

We tested the two extension module configuration, described in the previous section, in an indoor WSN deployment located at the basement of our department. The testbed setup (figure 6.5), consisted in 10 IRIS wireless devices [46] building a multi-hop WSN together with the MMS. Each IRIS mote was configured to generate one packet of 88 bytes every minute and to transmit it over the network. The MMS was set to locally sample one channel from each extension module generating 2 sensor readings every minute in order to transmit the sampled data over the wireless medium at the same rate as the IRIS motes. All data packets are routed through the multi-hop network toward a gateway, which is responsible for data aggregation and storing. We used the Collection Tree Protocol [56] as routing algorithm and the BoX-MAC medium access control protocol [83], both provided by the TinyOS operating system. To save energy, the MAC protocol was set with a radio duty cycle of 2% which represents a common value when long lasting WSNs are deployed. The testbed ran for 5 hours and collected statistical information embedded in the packets transmitted every minute by each node.

Table 6.3 reports detailed power consumption statistics, averaged over all the IRIS motes and compared with the one of the MMS. We observe that the power consumption of the MMS is similar to the one of the IRIS motes. Despite these measurements strongly depend on the position of the MMS in the network topology, this similarity shows that our solution is comparable in terms of energy requirements to a common WSN platform. Furthermore, the energy consumption of the sampling activity is less than 1% of the overall consumption that is largely dominated by the radio activity. We are aware that different transducers (e.g., the current-loop family) can consume more energy during conversion. This would significantly impact the fraction of energy consumed by the sampling activity. However, this consumption is exclusively related to the adopted transducer and does not depend on our architecture.

Table 6.3. Testbed - consumption details.

State	MMS Platform	IRIS Platform (AVG)
Energy Sleep (J)	0.26	0.13
Energy Radio Tx (J)	4.56	4.79
Energy Radio Rx (J)	1.23	1.28
Energy Radio Idle (J)	19.84	25.65
Energy Sampling (J)	0.23	0
Total	26.13	31.85

6.5.3 Design Overhead

As discussed in subsection 6.3, we adopted a new design for the low-power shared bus, increasing the flexibility and reducing the cost of the MMS. However, these advantages come at the expense of a higher management complexity of the system that caused an increase of the power consumption. Obviously, the higher is the number of extension modules, the higher is the complexity,

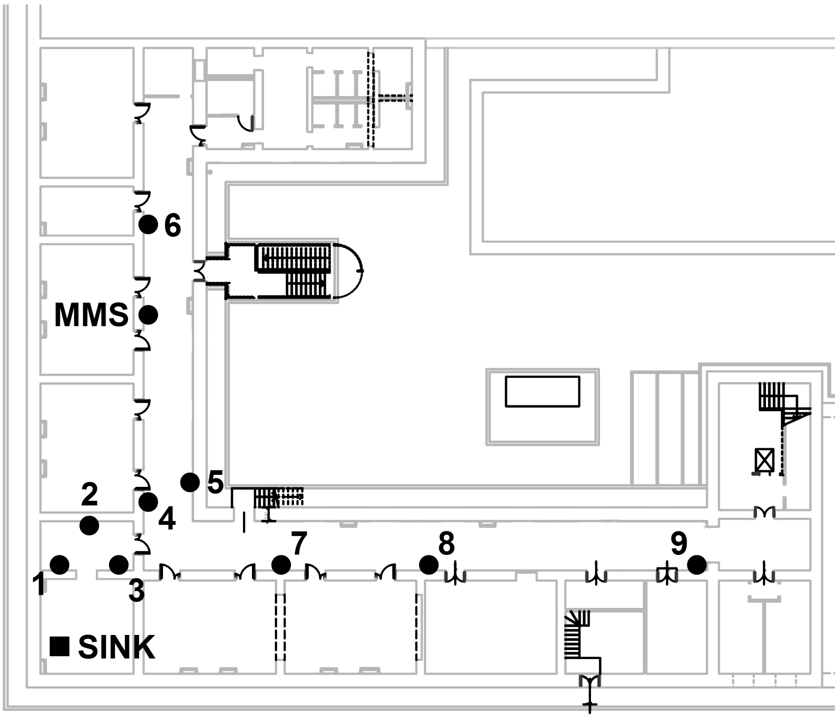


Figure 6.5. Testbed map.

thus the power consumption overhead increases accordingly. In this subsection, we evaluate the overhead of the new MMS design in terms of power consumption when compared to the original design. *Wake-up*, *polling* and *latency* are the three main factors that cause the consumption overhead. We consider a configuration made of a master module and 4 extension modules performing a local measurement cycle as the one described in section 6.5.1 to measure the contribution of each of those factors to the overall power consumption.

Wake-up. When the master module starts an I^2C communication, it writes on the bus the address of the module it wants to communicate with. This activity wakes up all the extension modules that start an address matching procedure. Thanks to the new architecture of the M0+ MCU of the extension modules, the address matching procedure is carried out only employing the I^2C peripheral integrated in the chip, without the need to awake the MCU (this mode is called *sleepwalking* [116]). This feature wakes up the MCU only upon an address match event, reducing the overall power consumption of the address matching procedure. At the standard I^2C 100 kbit/s datarate, the address transmission requires 90 μs. Assuming 3.3 V and 50 μA current consumption of the sleepwalking peripheral, the energy required by an extension module to perform an address matching procedure on the low-power shared bus is 14 nJ. Considering that for each request there are 3 extension modules that do not match the address, the overall energy wasted for each request is 42 nJ.

Polling. The master module polls each extension module by looking at the data flag area which notifies whenever a new data is available. On the low-power shared bus this operation is performed as in figure 6.6: the master first send a write request followed by the register address of the data flag area and successively a read request. The extension module replies with the content of the data flag area and, if no data is available, the communication ends. The whole procedure, which is 8 bytes long, is performed in 840 μs on a standard 100 kbit/s datarate I^2C channel. The measured current consumption of the MMS during a polling procedure is 4 mA that, assuming 3.3 V supply, brings to an energy of 10.1 μJ per extension module. In the configuration under test, there are 3 polled extension modules over 4 that do not provide new samples, hence, the wasted energy is 30.3 μJ .

Latency. We observed that the new design of the low-power shared bus requires, on average, 6 additional control bits each 10 data bits transmitted with the old implementation. This constraint increases latency in communications and, in turn, power consumption. The reason why more control bits are needed relies on the more complex I^2C bus management compared to the SPI protocol used by the old design. Assuming a local data measurement procedure as the one described in subsection 6.5.1, the protocol transmits 6 additional control bytes. Considering 4 mA current consumption of I^2C communication, 3.3 V supply and a standard 100 kbit/s I^2C datarate, the additional energy for each extension module measurement introduced by latency is 5.8 μJ. Note that we are not considering the latency introduced by the different supported data rates of each protocol since it is limited by noise immunity requirements.

Based on these measurements, the overall energy overhead of the low power shared bus in a local measurement procedure is 36.14 μJ which represents a consumption overhead of 9% when compared to the overall energy consumption measured in a single extension module configuration of table 6.2. Recalling that the sampling activity accounts for less than 1% of the overall energy consumption of the MMS, we can conclude that the energy overhead of the new MMS design is totally negligible.

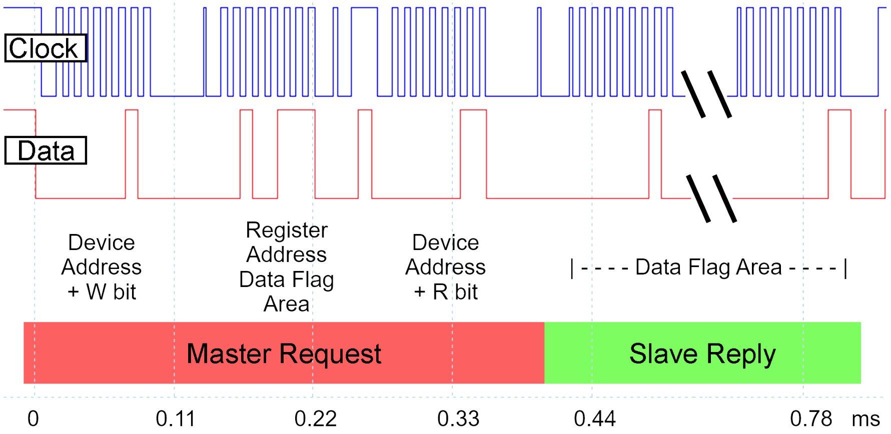


Figure 6.6. Polling I^2C communication detail.

6.6 Concluding Remarks

In this chapter, we presented further developments of the MMS, a novel low-power wireless modular architecture designed for SHM applications. The MMS takes advantage of a low-power shared bus connecting slotted extension modules that interact with a master in a master/slave communication abstraction. The extension modules, which can be combined as needed, allows the MMS to face the continuously evolving needs of most SHM scenarios. Thanks to its peculiar characteristics, the MMS overcomes commercial state-of-the-art WSN solutions for SHM, like the NI WSN, which do not offer enough flexibility to fulfill requirements of many application contexts in a both cost-effective and efficient way.

The new implementation overcomes the size and costs issues of the first prototype presented in [113] offering an improved architecture based on DIN standard modular housing. This solution also offers an increased flexibility by supporting interchangeable power sources. We validated the effectiveness of the system low-power design, performing energy measurements during data acquisition from actual transducers. In addition, we tested the MMS within a real WSN deployment, to demonstrate the compliance of the system in such applications.

In the near future, we plan the development of several additional extension modules able to support most of the sensors used in SHM, such as strain wire gauges, crack meters, inclinometers, displacement sensors, etc., and additional master modules to support different wireless frequencies (e.g., 433 MHz, 868 MHz, 915 MHz) and certified industrial wireless protocols like Wireless Hart.

Conclusion and Future Work

This thesis presented solutions to different problems emerging from the lack of energy availability in resource constrained sensor networks. We organized the thesis in two parts.

In part I we investigated solutions to the energy problem in wireless sensor networks (WSNs) approaching it from different perspectives that are summarized in figure 6.7. First of all, we proposed hardware solutions to the energy problem in WSNs and we outlined how the exploitation of low-power component is a necessary, but not sufficient, condition for the energy efficiency of motes (see chapter 1). Then, in chapter 2, we pointed out that the energy awareness of software layer runing on a mote is one of the most effective solution to save energy and we introduced two energy aware protocol we designed and implemented. Furthermore, we investigated the energy harvesting topic in relation to a new paradigm of energy exploitation in WSNs: energy neutrality (see chapter 3). A energy neutral mote is able to use the harvested energy at a suitable rate, allowing it to operate perennially. This approach, which is so different from the traditional mote lifetime maximization, raises new questions and issues new challenges. To face some of them we presented a new low power platform featuring hardware and software modules able to fulfill the needs of energy neutral applications (see section 3.1).

In part II of the thesis we applied the solutions presented in part I to structural health monitoring (SHM). In the specific, we introduced the WSN we designed to monitor the construction site of the Rome B1 underground (see chapter 4) and to gather significant data about damaged building and infrastructures in earthquake-stricken areas (see chapter 5).

Part I. In Wireless Sensor Networks the problem of the lack of energy is mainly due to the limited amount of energy available in batteries with respect to the needs of power hungry hardware modules, radio transceivers in particular. This translates in a reduction of network lifetime which prevents the full exploitation of WSNs in various application contexts. For such reasons an active research field aimed to mitigate the *energy problem* in WSNs raises in the last decade. We investigated this field and in this thesis we presented our solutions trying to prologue motes lifetime.

Figure 6.7 summarizes the contributions of this thesis to the *energy problem* in terms of low-power hardware, energy aware software, energy neutral platforms and layer where the proposed solutions are applied.

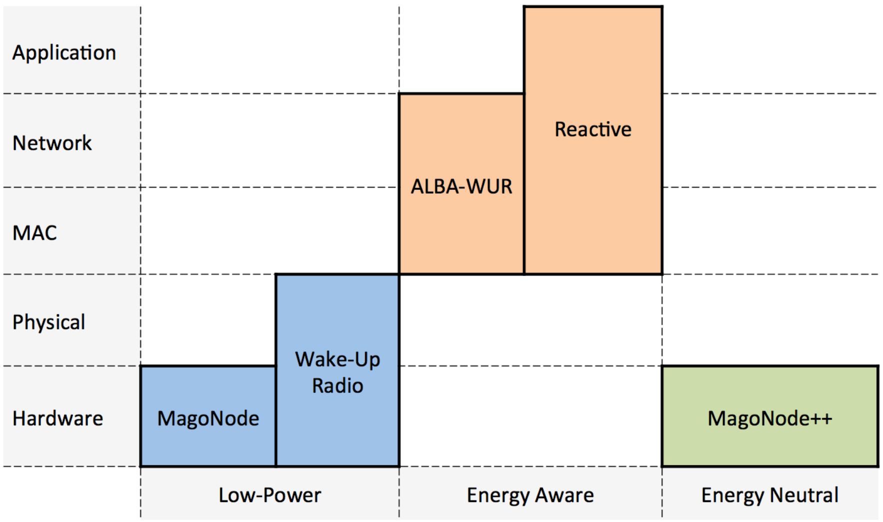


Figure 6.7. Thesis Contributions.

Low-power Hardware. In the first column of figure 6.7 are reported solutions to the energy problem based on low-power hardware we introduced in chapter 1. Starting from a solid state-of-art, we design and implemented a new low-power mote, the MagoNode (see section 1.1). It is equipped with an energy efficient radio front-end which strengthen communications improving channel reliability and radio range. We showed that the MagoNode performance in terms of current consumption is better than other motes featuring a radio front-end. Moreover, we showed that the advantages introduced by the MagoNode RF front-end help to contain packet retransmissions. In our testbed, this translates into energy savings with respect to unamplified motes, where the waste of power is driven by communications overhead.

It is proven that the idle listening of mote transceiver is the main source of energy waste in WSNs [26]. To completely eliminate the waste of energy due to idle listening, research has recently shifted towards communications architectures based on wake-up radios (WURs) (see section 1.2). A wake-up receiver (WRx) is a ultra low-power hardware module which continuously listen to the channel and wakes the mote main transceiver only when needed. In this thesis we introduced and detailed our WUR system made of an almost passive WRx, consuming only 579 nA while in idle listening. Our WRx is not only able to wake up the mote main transceiver completely avoiding its idle listening, but it can also wake up only specific motes on the basis of their address. To strengthen the robustness of WUR communications, we deeply investigated the the nature of noise perceived by our WRx, which adversely affects performance in terms of reliability, radio range and power consumption. Studying data gathered during noise characterization tests, we came out with hardware and software mechanisms able to mitigate the impact of noise on WUR communication. In the specific, we equipped the WRx with an hardware preamble detector able to filter out unwanted awakenings. On the software side, we decided to implement an oversampling decoding routine which is able to filter out shot-noise and, to face high noise environment, a FEC mechanism based on Hamming code. We performed several test to determine the effectiveness of the proposed solution. The outcome showed that our WUR system can be utilized in environment where the Bit Error Rate (BER) induced by noise is up to 10^{-2}, vice versa, when the BER induced by noise is in the order of 10^{-3} or below, it is not worth to use any Forward Error Correction (FEC) mechanism since it does not introduce any advantages compared to uncoded data. The advantages of wake-up radio in terms of energy saving come at the price of shorter radio range with respect to traditional active radio transceivers. This implies that a wake-up enabled mote is able to communicate with other nodes within tens of meters exploiting WUR, while it is able to reach nodes hundreds of meters away using the main transceiver. To overcome this issue it is possible to intervene on the software side, adapting existing communication protocols for WSN to support WUR technology. Another way to solve this drawback is to design new communication protocols which natively supports WUR. Even if the amount of work required to design from scratches or re-engineer a communication protocol to support WUR is not negligible, it is mandatory in order to integrate WUR in WSNs in a way which is relevant from the point of view of applications.

Energy Aware. In chapter 2 we introduced the research work done on energy aware solutions. An energy aware software layer is able to optimize the energy usage to accomplish typical WSN tasks (transmit, receive, sense, etc.). In this thesis we presented two energy aware protocols: REACTIVE and ALBA-WUR (see figure 6.7, second column) .

REACTIVE is a low-power over-the-air programming (OAP) protocol we implemented to improve the energy efficiency and lower the image dissemination time of Deluge T2, a well-known OAP protocol implemented in TinyOS. OAP is an energy consuming task since the dissemination of a new image is based on broadcast. In such context the use of an underlying low-power MAC protocol is mandatory to contain the energy overhead of radio communications. Anyway, such MAC protocols increase the overall network latency which degrades the dissemination time of OAP protocols. In section 2.1, we deeply analyzed the integration between Deluge T2 and a low-power MAC protocol featuring Low Power Listening (LPL) named BoX-MAC [83]. The results of this analysis is REACTIVE: a simple algorithm which is able to dynamically disable part of the LPL features to boost Deluge over-the-air reprogramming keeping a high energy efficiency. The results of our experiments, presented in section 2.1.4, showed that the image dissemination time is 7 times smaller than Deluge with LPL, while the energy consumption drops 2.6 times.

ALBA-WUR is a cross-layer protocol for data gathering in WSN that redesigns a leading routing protocol, ALBA-R [29], extending it to exploit the features of the wake-up radio 1.2. In section 2.2.3 we compared ALBA-WUR and ALBA-R via simulations on GreenCastalia [88] software. Performance results in WSN scenarios with increasing traffic showed impressive performance gains of ALBA-WUR over ALBA-R. In particular, nodes running ALBA-WUR in typical WSN application scenarios were able to delivery data to the sink by spending 300 times less energy than nodes with duty cycle, even if the duty cycle is very low. In networks with longer duty cycles ($d \geqslant 0.1$), ALBA-WUR achieved even greater improvements, obtaining energy savings ranging from two to five orders of magnitude. This translates in estimated network lifetimes that are decades longer than those achievable in networks with duty cycles. Moreover, results concerning latency showed that the use of WRx technology is effective in reducing the time needed to deliver packets to the sink.

Energy Neutrality. In chapter 3 we pointed out that a naïve way of using harvested energy is to treat it as a supplement of the battery energy and we explored a new paradigm of energy exploitation in WSNs: energy neutrality. A mote reaches the energy neutral mode of operation when it is able to use the available energy at a suitable rate which allows everlasting lifetime [12]. Energy neutrality opened a new research branch in WSNs which raises new issues that need to be addressed for a full exploitation of such paradigm. In this thesis we tried to address some of them from a hardware perspective. In particular, we stressed that maximizing energy efficiency of a hardware platform dedicated to WSNs is the key to reach energy neutral operation (ENO), still providing reasonable data rates and delays. So we designed and implemented a new energy neutral hardware platform dedicated to energy neutral applications endowed with our wake-up radio (WUR) system, the MagoNode++ (see figure 6.7, third column). In section 3.1.1, we presented MagoNode++ architecture which is made of the following hardware modules: our wake-up radio system for energy efficient wireless communications, light and thermoelectric harvesters to gather energy from the environment and a battery and power management module for long-lasting operations. Then, we showed how, orchestrating features provided by those hardware elements, it is possible to support self-sustainable operation and so energy neutrality. Finally, in section 3.1.2, we ran a series of experiments aimed to assess MagoNode++ energy consumption. Result showed that the MagoNode++ consumes only 2.8 μA in Low Power Mode with its WRx module in listening mode, outperforming existing prototypes such as the on presented in [30], with a reduction of almost 50% in energy consumption.

Part II. While carrying on our research work on solutions trying to mitigate the *energy problem*, we also faced a challenging real WSNs applicative context: structural health monitoring (SHM). Dealing with it helped us to better understand issues related to energy waste in WSNs and inspired some of the solutions to mitigate the effects of the *energy problem* in we presented earlier in this thesis. In part II we presented our solutions dedicated to SHM exploiting WSN.

In chapter 4 we presented the WSN we designed to monitor the construction site of Rome B1 underground. The network, made up of 32 CM5000 wireless motes [106] and a sink node, operated for about three months during the digging of a 700 m tunnel connecting two stops. During this period the network collected data on the health of concrete blocks composing the tunnel through the means of several strain sensors connected to the motes. To efficiently relay data to the sink, we used an ultra-low power protocol specifically designed for periodic data collection named DISSense [25]. Experimental data collected during the network operation showed an average Data Delivery Ratio (DDR) higher than 95% and an average radio duty cycle of motes of 0,22%.

In chapter 5 we introduced our WSN solutions able to gather data in earthquake-stricken areas. The work done was part of the research project named "TETRis - TETRA Innovative Open Source Services" [34]. The main goal of TETRis was to extend public safety services and applications, running on top of the TETRA [35], with data collected by WSNs deployed on field. Our WSN has been designed to monitor damaged buildings, to acquire information about the quality of the air and to relay data to a Smart Object, which in turn forwards information to a control point using the TETRA network [35]. Application requirements, provided by field experts, gave us requirements to face the considered scenario.Bearing in mind application constraints, we decided to based our WSN on the MagoNode platform (see section 1.1). To prove the effectiveness of our design choices, we deployed our network in an hybrid indoor/outdoor testbed that consistent with the reference TETRis scenario. The results of the experimental activity, reported in section 5.3, confirmed that WSNs can be effectively used to support the management of critical situations in earthquake-stricken areas.

Inspired by above cited on-field experiences, which allowed us to understand some peculiar aspects related to SHM, we decided to design our own low-power WSN platform dedicated to SHM: the Modular Monitoring System (MMS) (see section 6). The key feature of the MMS, as the name suggest, is the modularity, which allows an easy customization of the platform depending on the number and the type of sensors required. We validated the effectiveness of the MMS low-power design performing energy measurements during data acquisition from actual transducers and testing the platform within a real WSN deployment (see section 6.5).

Future Work. As we pointed out in 2.2.2, the ALBA-WUR protocol performance was validated through simulations. In the near future we foreseen to setup an indoor/outdoor testbed in order to assess protocol performance in terms of power consumption, packet delivery ratio and average packet latency. To do so we will deploy a network made of MagoNode equipped with out wake-up radio system and programmed with ALBA-WUR implementation for TinyOS we developed. We will compare testbed result with other on-field routing protocols performance.

In the scope of energy neutrality, we are designing a low-power routing protocol which is able to support energy neutral operation (ENO) of the network exploiting WUR technology. The Energy Neutral Routing Protocol based on Wake-up Radio (ENWU) is based on the idea that network nodes exchange information about their energy level in a very efficient way using wake-up radio. From energy level it is possible to obtain the quality of service (QoS) a node can support at given time, which is computed in number of packets a node can relay. Nodes energy level information is exploited by the ENWU to guarantee the everlasting operation of the network. Indeed, routing of packets would be done only through route that can support the required QoS. ENWU protocol is based on the assumption that the energy consumption of an almost passive wake-up receiver (WRx) can be perennially supported by a mote with harvesting capability since it is less than battery auto-discharge. The ENWU protocol will be compatible with TinyOS and, at low level, it will be supported by our MagoNode++. Indeed a radio wake-up enabled WSN hardware platform, featuring energy harvesting capabilities and able to estimate currently available energy, like the MagoNode++ is mandatory to carry out ENWU protocol operations.

In chapter 6 we presented the Modular Monitoring System (MMS) pointing out its strengths. In particular, we stressed how the MMS, taking advantage of a low-power shared bus, is able to connect slotted extension modules and orchestrate them through a master module. The extension modules, which can be combined as needed, allows the MMS to face the continuously evolving needs of most SHM scenarios. Right now only a general-purpose extension module, we designed as a demonstrator of the functionalities of our system, is available. However, in the near future, we are planning to develop several additional extension modules able to support most of the sensors used in SHM, such as strain wire gauges, crack meters, inclinometers, displacement sensors, etc., and additional master modules to support different wireless frequencies (e.g., 433 MHz, 868 MHz, 915 MHz) and certified industrial wireless protocols like Wireless Hart.

Bibliography

[1] Luca Mainetti, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1–6. IEEE, 2011. 1

[2] Gerd Kortuem, Fahim Kawsar, Daniel Fitton, and Vasughi Sundramoorthy. Smart objects as building blocks for the internet of things. *Internet Computing, IEEE*, 14(1):44–51, 2010. 1, 2

[3] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, pages 257–260. IEEE, 2012. 1

[4] Libelium. http://www.libelium.com. Accessed: 2016-05-27. 2

[5] Delphine Christin, Andreas Reinhardt, Parag S Mogre, and Ralf Steinmetz. Wireless sensor networks and the internet of things: selected challenges. *Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*, pages 31–34, 2009. 1

[6] Elyes Ben Hamida and Guillaume Chelius. Strategies for data dissemination to mobile sinks in wireless sensor networks. *IEEE Wireless Communications*, 15(6):31–37, 2008. 1

[7] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011. 1

[8] Carsten Bormann, Angelo P Castellani, and Zach Shelby. Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62, 2012. 1

[9] A. P. Chandrakasan and R. W. Brodersen. Low Power CMOS Digital Design. *Kluwer Academic Publishers, MA*, 1996. 3, 13

[10] M. Pedram and J. Rabaey. Power Aware Design Methodologies. *Kluwer Academic Publishers, MA*, 2002. 3, 13

[11] Sudevalayam, Sujesha and Kulkarni, Purushottam. Energy harvesting sensor nodes: Survey and implications. *Communications Surveys and Tutorials, IEEE*, 2011. 3, 4, 13, 87

[12] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6(4):32, 2007. 3, 4, 13, 87, 88, 89, 128

[13] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364–369. IEEE, 2005. 3, 13

[14] I.F. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 2002. 3, 4, 13

[15] B.P.S. Sahoo, Satyajit Rath and Deepak Puthal. Energy Efficient Protocols for Wireless Sensor Networks: A Survey and Approach. *International Journal of Computer Applications*, 2012. 3, 4, 13

[16] Kemal Akkaya and Mohamed Younis. A Survey on Routing Protocols for Wireless Sensor Networks. *Ad Hoc Networks - Elsevier B.V*, 2003. 3, 4, 13

[17] Langendoen, Koen and Meier, Andreas. Analyzing MAC Protocols for Low Data-rate Applications. *ACM Trans. Sen. Netw.*, 2010. 3, 4, 13, 66

[18] Nicolas Burri, Pascal Von Rickenbach, and Roger Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 450–459. IEEE, 2007. 3, 4, 13

[19] Răzvan Musăloiu-E, Chieh-Jan Mike Liang, and Andreas Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pages 421–432. IEEE, 2008. 3, 4, 13

[20] Elena Fasolo, Michele Rossi, Jörg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2):70–87, 2007. 3, 4, 13

[21] Winston K. G. Seah and 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. Wireless VITAE 2009*, 2009. 3, 4, 13, 87

[22] Alippi, C. and Galperti, C. An Adaptive System for Optimal Solar Energy Harvesting in Wireless Sensor Network Nodes. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 2008. 3, 4, 13, 87

[23] Chee, Yuen-Hui and Koplow, Mike and Mark, Michael and Pletcher, Nathan and Seeman, Mike and Burghardt, Fred and Steingart, Dan and Rabaey, Jan and Wright, Paul and Sanders, Seth. PicoCube a 1 Cm3 Sensor Node Powered by Harvested Energy. *Proceedings of the 45th Annual Design Automation Conference*, 2008. 3, 4, 13, 87

[24] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman and M. B. Srivastava. Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. *IPSN*, 2005. 3, 4, 13, 87, 92

[25] Ugo Maria Colesanti, Silvia Santini, and Andrea Vitaletti. Dissense: An adaptive ultralow-power communication protocol for wireless sensor networks. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–10. IEEE, 2011. 4, 8, 98, 129

[26] Ilker Demirkol, Cem Ersoy, Fatih Alagoz, et al. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006. 6, 16, 126

[27] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005. 7, 63

[28] David Moss and Philip Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, pages 116–119, 2008. 7, 116

[29] Chiara Petrioli, Michele Nati, Paolo Casari, Michele Zorzi, and Stefano Basagni. Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 25(3):529–539, 2014. 7, 63, 75, 77, 128

[30] Trong Nhan Le, Michele Magno, Alain Pegatoquet, Olivier Berder, Olivier Sentieys, and Emanuel Popovici. Ultra low power asynchronous mac protocol using wake-up radio for energy neutral wsn. In *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*, pages 10–15. ACM, 2013. 8, 88, 92, 129

[31] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 13–24, New York, NY, USA, 2004. ACM. 8, 95, 110

[32] Jerome P. Lynch and Kenneth J. Loh. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, 38(2):91–130, 2006. 8, 95, 110

[33] Tyler Harms, Sahra Sedigh, and Filippo Bastianini. Structural health monitoring of bridges using wireless sensor networks. *Instrumentation & Measurement Magazine, IEEE*, 13(6):14–18, 2010. 8, 95, 110

[34] TETRis Project. http://www.ponrec.it/open-data/risultati/ricerca-industriale/pon01_00451/. Accessed: 2016-03-01. 9, 101, 129

[35] TETRA. http://www.tandcca.com/. Accessed: 2015-09-30. 9, 101, 129

[36] IEEE 802.15.4. https://standards.ieee.org/. 17

[37] Iris, TelosB and Micaz motes. http://www.memsic.com/wireless-sensor-networks. 17

[38] UCMote mini. http://www.ucmote.com. 17, 21

[39] ETSI EN 300 328 V1.8.1 final draft (2012-04). http://www.etsi.org. 17, 18

[40] FCC CFR 47. https://www.fcc.gov/. 17, 19

[41] ZigBee Alliance. http://www.zigbee.org/. 17

[42] Genesi Project Website. http://genesi.di.uniroma1.it. 17, 97, 100, 110

[43] ZigBit ATZB-A24-UFL/U0 datasheet. http://www.atmel.com. 17, 20

[44] Dresden Elektronik OEM modules deRFmega. http://www.dresden-elektronik.de. 17, 20

[45] WSense s.r.l. http://www.wsense.it/. 18

[46] Memsic website - IRIS mote. http://www.memsic.com/wireless-sensor-networks/. Accessed: 2015-11-06. 18, 119

[47] Zhang Hao, Li Zhiqun, Zhang Meng, and Chen Gang. A 2.4 ghz low-if rf frontend for wireless sensor networks. In *Microwave and Millimeter Wave Technology (ICMMT), 2010 International Conference on*, pages 225–228. IEEE, 2010. 18

[48] Amine Didioui, Carolynn Bernier, Dominique Morche, and Olivier Sentieys. Impact of rf front-end nonlinearity on wsn communications. In *Wireless Communication Systems (ISWCS), 2012 International Symposium on*, pages 875–879. IEEE, 2012. 18

[49] Ilker Demirkol, Cem Ersoy, Fatih Alagoz, et al. Mac protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006. 18, 19

[50] Texas Instruments Application Note AN086. http://www.ti.com. 18, 19

[51] AdvanticSYS CM3300 datasheet. http://www.advanticsys.com. 20

[52] TinyOS. http://www.tinyos.net/. 20, 24, 25, 64, 67, 90, 98, 116

[53] Contiki. http://www.contiki-os.org. 20, 24, 65

[54] Texas Instruments Application Note AN098. http://www.ti.com. 22

[55] David Moss, Jonathan Hui, and Kevin Klues. Low power listening. *TinyOS Core Working Group, TEP*, 105, 2007. 24, 27, 103

[56] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pages 1–14. ACM, 2009. 24, 27, 67, 103, 116, 120

[57] Ugo Colesanti and Silvia Santini. The collection tree protocol for the castalia wireless sensor networks simulator. *ETH Zurich, Zurich, Switzerland*, 2011. 24, 27

[58] Daquan Feng; Chenzi Jiang; Gubong Lim; Cimini, L.J., Jr.; Gang Feng; Li, G.Y. A survey of energy-efficient wireless communications. *Communications Surveys and Tutorials, IEEE*, 2013. 32

[59] R. Jurdak, P. Baldi, and C. Videira Lopes. Adaptive Low Power Listening for Wireless Sensor Networks. *In IEEE Trans. on Mobile Comp. 2007*, 2007. 32

[60] L. Gu and J. A. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems*, 29(2–3):157–182, 2005. 33

[61] L. Chen, S. Cool, H. Ba, W. Heinzelman, I. Demirkol, U. Muncuk, K. R. Chowdhury, and S. Basagni. Range extension of passive wake-up radio systems through energy harvesting. In *Proceedings of IEEE ICC 2013*, pages 142–147, Budapest, Hungary, June 9–13 2013. 33

[62] L. Chen, J. Warner, P. L. Yung, D. Zhou, W. Heinzelman, I. Demirkol, U. Muncuk, K. Chowdhury, and S. Basagni. REACH^2-Mote: A Range Extending Passive Wake-up Wireless Sensor Node. *ACM Transactions on Sensor Networks*, 11(4):64:1–64:33, December 2015. 33

[63] EPCglobal, Inc. EPC$^{\text{TM}}$ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz, ver.1.1.0, 2006. 33

[64] J.-P. Curty, N. Joehl, C. Dehollaini, and M. J. Declercq. Remotely powered addressable UHF RFID integrated system. *IEEE Journal of Solid-State Circuits*, 40(11):2193–2202, November 2005. 33

[65] E. Nilsson and C. Svensson. Ultra low power wake-up radio using envelope detector and transmission line voltage transformer. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(1):5–12, March 2013. 34

[66] Joaquim Oller, Ilker Demirkol, Jordi Casademont, and Josep Paradells. Design, development, and performance evaluation of a low-cost, low-power wake-up radio system for wireless sensor networks. *ACM Transactions on Sensor Networks*, 10(1):11:1–11:24, December 2013. 34

[67] A. S. Boaventura and N. B. Carvalho. A low-power wake-up radio for application in WSN-based indoor location systems. *International Journal of Wireless Information Networks*, 20(1):67–73, 2013. 34

[68] N. E. Roberts and D. D. Wentzloff. A 98 nW wake-up radio for wireless body area networks. In *Proceedings of IEEE RFIC, 2012*, pages 373–376, Montreal, Canada, June 17–19 2012. 34

[69] S. J. Marinkovic and E. M. Popovici. Nano-power wireless wake-up receiver with serial peripheral interface. *IEEE Journal on Selected Areas in Communications*, 29(8):1641–1647, September 2011. 34

[70] Texas Instruments. http://www.ti.com. Accessed: 2015-03-22. 35

[71] Michele Magno and Luca Benini. An ultra low power high sensitivity wake-up radio receiver with addressing capability. In *Proceedings of GROWN Workshop 2014*, pages 92–99, Oct 2014. 36

[72] ITU Radio Regulations. http://www.itu.int/pub/R-REG-RR. Accessed: 2015-10-12. 39

[73] HyungJune Lee, Alberto Cerpa, and Philip Levis. Improving wireless simulation through noise modeling. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 21–30. IEEE, 2007. 41

[74] Nicholas M Boers, Ioanis Nikolaidis, and Pawel Gburzynski. Patterns in the rssi traces from an indoor urban environment. In *Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD), 2010 15th IEEE International Workshop on*, pages 61–65. IEEE, 2010. 41

[75] EMSPCC11 mote. http://olsonet.com/Documents/emspcc11.pdf. Accessed: 2015-11-03. 41

[76] Brian P Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T Sakai. Ieee 802.11 wireless local area networks. *Communications Magazine, IEEE*, 35(9):116–126, 1997. 42

[77] Stephen B Wicker. *Error control systems for digital communication and storage*, volume 1. Prentice hall Englewood Cliffs, 1995. 50

[78] Roberto Cusani and Tiziano Inzerilli. *Teoria dell'informazione e codici.* Ingegneria 2000, 2008. 50

[79] Panta, Rajesh Krishna and Bagchi, Saurabh and Midkiff, Samuel P. Zephyr: Efficient incremental reprogramming of sensor nodes using function call indirections and difference computation. *Proc. of USENIX Annual Technical Conference*, 2009. 64, 65

[80] Munawar, W. and Alizai, M.H. and Landsiedel, O. and Wehrle, K. Dynamic TinyOS: Modular and Transparent Incremental Code-Updates for Sensor Networks. *Communications (ICC), 2010 IEEE International Conference on*, 2010. 64, 65

[81] Dong, Wei and Liu, Yunhao and Wu, Xiaofan and Gu, Lin and Chen, Chun. Elon: enabling efficient and long-term reprogramming for wireless sensor networks. *ACM SIGMETRICS Performance Evaluation Review*, 2010. 64, 65, 66

[82] Panta, R.K. and Khalil, I. and Bagchi, S. Stream: Low Overhead Wireless Reprogramming for Sensor Networks. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007. 64, 65

[83] Moss, David and Levis, Philip. BoX-MACs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, 2008. 64, 66, 120, 128

[84] Jonathan W. Hui and David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. pages 81–94, 2004. 67

[85] Gilman Tolle and David E. Culler. Design of an application-cooperative management system for wireless sensor networks. pages 121–132, 2005. 67

[86] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, pages 2–2, Berkeley, CA, USA, 2004. USENIX Association. 67

[87] Vana Jelicic, Michele Magno, Davide Brunelli, Vedran Bilas, and Luca Benini. Analytic comparison of wake-up receivers for wsns and benefits over the wake-on radio scheme. In *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pages 99–106. ACM, 2012. 75

[88] David Benedetti, Chiara Petrioli, and Dora Spenza. Greencastalia: an energy-harvesting-enabled framework for the castalia simulator. In *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*, page 7. ACM, 2013. 75, 81, 128

[89] Athanassios Boulis. Castalia: revealing pitfalls in designing distributed algorithms in wsn. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 407–408. ACM, 2007. 76, 81

[90] Thanos Stathopoulos, Martin Lukac, D Mclntire, John Heidemann, Deborah Estrin, and William J Kaiser. End-to-end routing for dual-radio sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2252–2260. IEEE, 2007. 76

[91] Stevan J Marinkovic and Emanuel M Popovici. Power efficient networking using a novel wake-up radio. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 139–143. IEEE, 2011. 77

[92] Johannes Blanckenstein, Jirka Klaue, and Holger Karl. Energy efficient clustering using a wake-up receiver. In *European Wireless, 2012. EW. 18th European Wireless Conference*, pages 1–8. VDE, 2012. 77

[93] Christian Hambeck, Stefan Mahlknecht, and Thomas Herndl. A 2.4 μw wake-up receiver for wireless sensor nodes with -71 dbm sensitivity. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 534–537. IEEE, 2011. 77

[94] Chiara Petrioli, Dora Spenza, Pasquale Tommasino, and Alessandro Trifiletti. A novel wake-up receiver with addressing capability for wireless sensor nodes. In *Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on*, pages 18–25. IEEE, 2014. 77

[95] Isidor Buchmann. Batteries in a portable world: A handbook on rechargeable batteries for non-engineers. *Richmond, Canada*, 2001. 85

[96] Chen, Wei and Andreopoulos, Yiannis and Wassell, Ian J and Rodrigues, Miguel RD. Unlocking Energy Neutrality in Energy Harvesting Wireless Sensor Networks: An Approach Based on Distributed Compressed Sensing. *arXiv preprint arXiv:1312.4207*, 2013. 88

[97] Chen, Wei and Andreopoulos, Yiannis and Wassell, Ian J and Rodrigues Miguel RD. Towards Energy Neutrality in Energy Harvesting Wireless Sensor Networks: A Case for Distributed Compressive Sensing? *arXiv preprint arXiv:1310.4761*, 2013. 88

[98] Castiglione, Paolo and Simeone, Osvaldo and Erkip, Elza and Zemen, Thomas. Energy-neutral source-channel coding in energy-harvesting wireless sensors. *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, 2011. 88

[99] Alessandro Bogliolo, Valerio Freschi, Emanuele Lattanzi, Amy L Murphy, and Usman Raza. Towards a true energetically sustainable wsn: A case study with prediction-based data collection and a wake-up receiver. In *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*, pages 21–28. IEEE, 2014. 88

[100] Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic. *Mobile Ad Hoc networking: the cutting edge directions*, volume 35. John Wiley & Sons, 2013. 89

[101] Solexperts website. http://www.solexperts.com/. Accessed: 2015-03-27. 95, 109, 110

[102] Geokon website. http://www.geokon.com/8002-16. Accessed: 2015-03-27. 95, 110

[103] Keller website. http://www.keller-druck.com/. Accessed: 2015-03-27. 95, 110

[104] NI WiFi-DAQ. www.ni.com/data-acquisition/wireless/. Accessed: 2015-03-30. 95, 110

[105] Microstrain WSN. http://www.microstrain.com/wireless. Accessed: 2015-03-27. 95, 110

[106] Cm5000 wireless mote. http://www.advanticsys.com. Accessed: 2013-11-11. 98, 129

[107] Raytech. http://www.raytech.it. Accessed: 2012-06-15. 98

[108] Treesse Engineering. http://www.treesseengineering.com. Accessed: 2012-06-16. 100

[109] Stefan Poslad. *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Wiley Publishing, 1st edition, 2009. 101

[110] SISGEO. http://www.sisgeo.com/. Accessed: 2014-03-14. 104

[111] National Instruments WSN. http://www.ni.com/wsn/i/. Accessed: 2015-03-27. 110

[112] Structural health monitoring at the pont de la poya, fribourg. In *White Paper, Genesi Project*, 2013. 110

[113] F. Angeletti, M. Paoli, U. M. Colesanti, and A. Vitaletti. Wireless sensor networks in structural health monitoring: A modular approach. In *SENSORCOMM*, page to be published. Iaria, 2015. 111, 116, 123

[114] PIC32MX Family Reference Manual, 2008. Microchip Technology inc., pp. 24-1 24-74. 113

[115] FieldComm Group website - Wireless HART. http://en.hartcomm.org/. Accessed: 2015-11-06. 116

[116] How to implement SleepWalking on an ARM Cortex-M4 MCU Application. Atmel application note AT04113. 121