Sapienza Università di Roma

Department of Computer Science

Doctoral Program in Computer Science (XXXI Cycle)

Real-time and long lasting Internet of Things through semantic wake up radios

Federico Ceccarelli

September 2019

Thesis commitee

Prof. Chiara Petrioli (advisor)
Department of Computer Science
Sapienza - University of Rome
Rome, Italy

Prof. Enrico Tronci
Department of Computer Science
Sapienza - University of Rome
Rome, Italy

Prof. Gaia Maselli
Department of Computer Science
Sapienza - University of Rome
Rome, Italy

External Reviewers

Prof. Cristian Cassella
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts, USA

Prof. Sergio Palazzo
Department of Electrical, Electronic and Computer Engineering
University of Catania
Catania, Italy

Abstract

The world is going towards the Internet of Things (IoT) where trillions of objects that are common in our lives will be enhanced and revolutionized by adding them computational and networking capabilities. Examples are cars, street lamps, industrial machinery, electrical appliances. The cornerstone of Internet of Things research is Wireless Sensor Networks (WSNs). These networks are made of hundreds of low-cost, low-complexity devices endowed with sensors to monitor the surrounding environment or objects. Typically these devices (also called sensors, nodes or motes) are battery-powered, therefore they can operate for a limited amount of time (i.e., days) before running out of energy. This is the main challenge that applications of Wireless Sensor Networks have to face.

Since one of the major power consumers in a node is the radio transceiver, a lot of research effort has been put into finding solutions that keep the radio in a low-power state as much as possible, while not harming the communication capability. While this approach brings the network lifetime, i.e. the time before battery-operated nodes die having depleted their energy, to years or more, it introduces significant latency, as the energy reduction comes at the cost of not being able to reach nodes in deep sleep for long period of times. The most promising solution to this problem is the wake-up radio, an additional ultra-low power transceiver used for the sole purpose of triggering the activation of the high power, high bandwidth radio. Wake-up radio enabled IoT systems maintain always on their wake up radio, which has a negligible energy consumption, in this way optimizing both energy and latency performance metrics. Most of the research so far focused on the design of wake-up receivers, while a limited amount of communication protocols that take advantage of this radio has been proposed. Moreover, almost all of these protocols have been evaluated only through simulations.

In this thesis we set to start filling this gap. We first evaluate the range performance of an ultra-low power wake-up receiver integrated into a state-of-the-art Wireless Sensor Network mote, the MagoNode++. Based on the results of this evaluation we deploy an outdoor testbed made of MagoNode++ motes. The testbed allows to validate in a real-world scenario our implementation of CTP-WUR, an extension of the widely used Collection Tree Protocol (CTP) for wake-up radio-enabled Wireless Sensor Networks. The comparison between CTP-WUR and CTP demonstrates that wake-up radios can effectively reduce the power consumption and obtain, at the same time, end-to-end latencies in the order of milliseconds, enabling new time critical applications. Based on the results and on the insights gained during the testbed evaluation a new version of CTP-WUR is presented that improves its performance across all the metrics taken into consideration: end-to-end packet latency, energy consumption and Packet Delivery Ratio.

Contents

Chapter 1

Introduction

The advancements of the electronics in the last decade have made possible the realization of radio transceivers, microcontrollers, memories and sensors small enough (yet powerful) to be embedded inside objects that surround us in the everyday life, paving the way for new applications that can enhance our quality of life [1]. This is the so-called Internet of Things (IoT). The Internet of Things appears in many different ways (e.g., Smart Cities or Industry 4.0) but one of the cornerstones is represented by Wireless Sensor Networks (WSNs): networks of low-cost and low-complexity devices, equipped with a variety of sensors and a radio transceiver that cooperate to measure some physical quantity in the real world. These devices (also called sensors, nodes or motes) create multi-hop, ad-hoc networks in order to forward the measured data towards one or more collection endpoints, the sinks.

Thanks to their flexibility they allow applications such as Structural Health Monitoring of critical infrastructures [2] or smart healthcare systems [3]. Most of these applications have to face a challenge: they are made of battery-powered devices, therefore the available amount of energy is limited. In general, frequently replacing the batteries has a remarkable impact on the costs and, in some case, can be impractical. An example is a wireless sensor network made of some tens of devices monitoring the stability of a bridge. This challenge constitutes a major obstacle to long-term deployments. For this reason, a lot of research has been done in order to increase the operational lifetime of wireless networks.

An approach to this problem is to harvest energy from the environment using small solar panels or wind turbines to recharge the battery or a super-capacitor [4, 2]. On the other hand, the intermittent nature of these sources of energy makes them impossible to rely on. Therefore, it is essential to try to reduce as much as possible the energy consumption of network nodes. The hardware component that draws the most energy is the radio transceiver, irrespective of its state, either transmitting or receiving. However, most of the time the radio is not busy in any of these two activities, but it listens on the channel for incoming transmissions. This is called *idle listening*. Since the traffic rate in this kind of networks is, in general, very low, keeping the radio active waiting for a transmission that may be minutes away is a waste of energy. Therefore, the goal is to keep it in a low-power mode as long as possible, where it consumes orders of magnitude less energy than in the active state. But, in the low-power state the radio is not able to receive anything.

Consequently there is a trade-off between energy consumption and communication capability. The established solution to this problem is to turn on and off periodically according to a pre-set *duty-cycle* the radio in order to allow communications only during certain time intervals. Several MAC protocols have been presented that follow this approach, each one trying to optimize some aspect. There are protocols, called synchronized [5, 6], where each network node exchanges with its neighbors its wake-up schedule so that it knows when each one of them will activate its radio and will be able to receive. Other protocols are asynchronous: each node knows only its own duty-cycle [7, 8, 6]. When it wants to transmit to a neighbor it has to "intercept" its active period. To do that it precedes the actual packet transmission with a preamble long enough to be heard by the recipient the next time it will turn on its radio. Several different optimizations on the length and format of the preamble have been proposed. This technique effectively reduces power consumption, allowing network to be operative for several months [9], but it has two drawbacks: i) it does not completely remove idle listening, which is only reduced, and ii) it increases packet latencies, because a node has to wait for the receiving node to activate its radio before actually transmitting the packet. On large deployments, the end-to-end packet latency can be in the order of *seconds*, making these protocols unfit for time critical applications (e.g., a WSN monitoring a bridge). It is clear that duty-cycling introduces a trade-off between energy efficiency and latency.

To remove this trade-off research has proposed the use of low-power triggering techniques, such as equipping network nodes with wake-up radios. A wake-up radio is an additional transceiver that flanks the main one, therefore called *main radio*. It is designed to have a consumption comparable to that of the main radio when this is in its lowest-power state (called *sleep state*). Typical values of current consumption of a main radio in an active state (i.e., transmitting, receiving or idle listening) are in the order of few tens of mA. These values go down to few μA when the radio is put in the sleep state [10, 11, 12]. With this approach the main radio is brought in the active state only when it has to transmit or receive a packet. Instead, the wake-up radio is always active waiting for a specific signal (the *wake-up signal*) to trigger the main radio activation.

Therefore the wake-up radio allows to completely remove idle listening and, at the same time, keeps latencies low, in the order of milliseconds [13, 14, 15].

In order to avoid that all the nodes inside the communication range of the wake-up radio activate their main transceiver, different addressing methods can be used. With identity-based addressing the wake-up signal contains the ID of the recipient. *Semantic addressing* allows to wake-up only the neighbors that have certain characteristics, described by metrics that are specific of the used protocol. Examples of metrics are the distance from the destination or the residual amount of energy of the node. Every network node has a pool of wake-up addresses that codify its status based on the chosen metrics. The wake-up signal contains the metrics values that a possible receiver should have.

So far the research on wake-up radio-enabled wireless sensor networking has focused mainly on the design of hardware that consumes as little energy as possible. In general, to obtain the required ultra-low consumptions sensitivity and bitrate are sacrificed. This means that a wake-up radio receiver has a shorter communication range and is slower than the main radio. New problems arise: for example routing paths are longer when wake-up radios are used. In recent years, the wireless networking research community have started to present more protocols taking into account the capabilities and shortcomings of wake-up radios. This prompted also the IEEE 802.11 Working Group to start the process of standardization of wake-up radios. The new standard, called IEEE 802.11ba [16] aims at defining the physical

(PHY) and Medium Access Control (MAC) layers of wake-up radio devices that are thought to be used together with traditional IEEE 802.11 devices in order to reduce their energy consumptions.

Almost all the protocols proposed so far have been evaluated through simulations. Only a few of them have been implemented and tested on real hardware. Simulations give many insights on the behavior of protocols, even on large scale networks, and allow to validate the ideas that underpin their design. But when protocols are tested in a real-world environment several new circumstances must be taken into account. To name a few of them: costs, coexistence with other wireless services (e.g., cellular networks, WLANs), hardware malfunctions, weather conditions. Therefore real testbeds available for wake-up radio-based networking are quite rare. An example is the Flocklab [17].

In this thesis we will present the results of our research on wake-up radio-based wireless sensor networks using a real-world testbed. In order to build the testbed, it has been necessary to evaluate the wake-up range attainable by the MagoNode++, a state-of-the-art mote for WSNs that features an ultra-low power wake-up receiver. Then we evaluate our implementation of CTP-WUR, a protocol designed to adapt the leading routing protocol for data collection in WSNs, the Collection Tree Protocol (CTP). Its comparison against CTP with duty-cycle proves that wake-up radio technology can effectively reduce energy consumptions and obtain end-to-end latencies in the order of milliseconds, paving the way for time critical applications. Thanks to the insights obtained through the testbed evaluation, we also propose, implement and validate an optimization of CTP-WUR that further reduces consumptions and latencies.

The rest of the thesis is organized as follows. In Chapter 2 we give a general overview to the state of the art of the research on wake-up radio-enabled WSNs.

In Chapter 3 we evaluate the range performance of an ultra-low power wake-up radio receiver (WuR) integrated into a wireless device suitable for wireless sensor networking deployments. We run several ranging experiments, both indoors and outdoors, where a transmitter sends wake-up sequences to a receiver positioned meters away. We measure the amount of received sequences and whether they incur errors or not. Our experiments show that for distances up to 24 meters indoors the tested WuR receives more than 96% of the transmitted sequences. The WuR performs slightly better outdoors, with more than 99% of the sequences being received with negligible amounts of errors.

In Chapter 4 we evaluate the effectiveness of wake-up radios in a real-world multi-hop testbed. To this end, we implement CTP-WUR, an extension of the Collection Tree Protocol (CTP), designed to take advantage of wake-up receivers. We also present a variation of CTP-WUR, that improves its performance. Both versions are compared, in our testbed, against CTP with duty cycle. The results of our experiments show that the use of wake-up radios produces remarkable energy savings (up to 15 times) and achieves latencies that are up to 10 times lower than those achieved by CTP with duty cycled radios.

1.1 List of publications

In the following the papers accepted for publications or submitted for review that form the basis of this thesis are listed.

- S. Basagni, F. Ceccarelli, F. Gattuso, C. Petrioli. "Demo Abstract: Abating LPL-induced Latency with Wake-up Radio Technology". In *Proceedings of ACM/IEEE IoTDI 2017, Pittsburgh, PA, USA. April 18 – 21 2017.*
- S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, A.V. Sheshashayee. "Wake-up Radio Ranges: A Performance Study". In *Proceedings of IEEE WCNC 2019. Marrakech, Morocco. April 15 – 19 2019.*
- A journal paper with the results described in this thesis is currently in preparation.

Chapter 2

State of the art

This section provides a brief overview of the research on wake-up radio-enabled Wireless Sensor Networks, covering both hardware designs and networking protocols. A broad survey of the field is presented by Piyare et al. in [18].

2.1 Wake-up radio design

Over the last twenty years several wake-up radio receivers prototypes have been presented. In this section we will describe some of them (summarized in Table 2.1).

Table 2.1: Wake-up Radio Prototypes

Authors	**Type**	**Freq. [MHz]**	**Sens. [dBm]**	**Pwr** [μW]
Gu et al. [19]	Passive	433	N/A	N/A
Nilsson et al. [20]	Passive	2400	-47	2.3
Oller et al. [21]	Active	868	-45	2.67
Spenza et al. [9]	Semi-active	868	-55	1.267
Petrioli et al. [22]	Active	2400	-83	1620
Gamm et al. [23]	Semi-active	868	-52	7.8
Sutton et al. [24]	Active	434	-52	8.1
Roberts et al. [25]	Semi-active	915	-41	0.098
Oh et al. [26]	Semi-active	402/915/2400	-43.2	0.116
Elgani et at. [27]	N/A	868	-55	0.013

The first design of a wake-up radio was presented by Gu and Stankovic in [19]: it is a passive circuit that harvests the power from the received radio signal to generate a wake-up interrupt. It is evaluated through simulations showing that it reaches a wake-up range of 3 meters, enough for short-range applications such as Wireless Body Area Networks (WBANs), but insufficient for the typical applications that require longer ranges. Another way to realize a completely passive wake-up receiver is using RF-powered RFID tags, as in [28] and [29]. They have a wake-up range of around 10 meters, a typical value for UHF RFID systems [30]. This kind of solutions, however, are not suited for applications that require peer-to-peer communications between the nodes. In fact, in order to transmit a wake-up sequence, a RFID reader is required, which has a power consumption in the order of Watts.

To have higher wake-up range, it is necessary to allow that some parts of the circuitry drain current from the battery. Typical solutions use Schottky diodes, MOSFETs or ad-hoc ICs to implement the envelope detector and a low-power comparator to generate the wake-up interrupt. In this way higher sensitivities can be obtained. For example, the design proposed by Nilsson et al. [20] reaches a -47 dBm sensitivity with a power consumption of $2.3\mu W$. Oller et al. [21] presented a wake-up receiver featuring a -45 dBm sensitivity and a current consumption lower than $1\mu A$; in their measurements the wake-up range was above 10 meters. Spenza et al. [9] presented a semi-active wake-up receiver with an ultra-low power consumption of $1.276\mu W$ and addressing capabilities. The wake-up signal is modulated using the On-Off Keying (OOK) scheme. This allows to keep the design of the receiver simple, which can be seen as made of four parts: the matching network, the envelope detector, the comparator (the only active component) and a preamble detector. An optional ultra-low power microcontroller can be used to perform address matching. The maximum sensitivity that has been measured is -55 dBm. The authors tested several data rates, from 1 kbps up to 100 kbps and measured the maximum wake-up range attainable: 45 meters when they considered only the capability of generating the wake-up interrupt and 32 meters for selective awakenings. Petrioli et al. in [22] presented a wake-up receiver prototype operating in the 2.4 GHz band, with selective addressing capability. In their design nodes are addressed using a sequence of continuous-wave or narrow-band pulses on a combination of some the 16 channels established by the IEEE 802.15.4 standard on the 2.4 GHz band.

This architecture has a sensitivity of -83 dBm, which allowed to obtain a wake-up range of 120 meters in outdoor in-field experiments. Its power consumption is $168\mu W$.

Paoli et al. [31] presented the MagoNode++, a WSN platform for Energy Neutral applications. It integrates the wake-up radio presented in [9] with a light or thermoelectric energy harvester and with a battery and power management module. Bedogni et al. [32] presents the architecture of a new Wake-up Radio IoT node. It uses a wake-up radio receiver designed by STMicroelectronics that operates in the 868 MHz frequency band. This receiver can be powered either by the node's battery (i.e., it is an active receiver) or by the RF signal (i.e., it is passive). In the first case it has a sensitivity of -38 dBm, while in the latter case it reaches -18 dBm. Moreover this architecture supports two operative modes: in the first one, called *Switch mode*, the node can be completely shut down using a hardware switch, reducing to zero the power consumption (plus the eventual power consumption of the wake-up receiver, if operating in an active way). This means that when the node receives a wake-up signal, it incurs in the additional overhead, both in terms of energy and time, of the boot phase. In the second operative mode, called *Low power mode*, the mote is put into a deep sleep state, hence continuously consuming a minimum amount of energy, and when it receives a wake-up signal an interrupt awakes the MCU moving it into an active state. These operative modes can be selected via software and the authors show how the choice on which one is best suited for an application, depends on how often a node should be activated during its operational lifetime.

There also exist off-the-shelf wake-up receivers, like the AS393X series from Austria Microsystems [33]. They are low-power low-frequency Amplitude Shift Keying (ASK) wake-up receivers which generate an interrupt when a signal at a carrier frequency between 110 kHz and 150 kHz is detected. They are equipped with correlators able to detect 16-bit or 32-bit programmable wake-up sequences. Therefore they can support semantic addressing. This family of receivers has a current consumption that varies from $1.37\mu A$ up to $2.7\mu A$ in listening mode and from $5.3\mu A$ to $8.3\mu A$ in active mode. They are used in the design of the receivers presented by Gamm et al. in [23] and Sutton et al. in [24]. The latter is also used to receive data packets. It has a power consumption of $8.1\mu W$ with a -52 dBm sensitivity that allowed ranges up to 15 meters in non-line-of-sight indoor experiments and up to 30 meters in line-of-sight outdoor experiments.

Some proposed designs feature nano-power consumptions. However they have very short wake-up ranges, that make them unfit for general sensing systems applications. An example is the receiver presented by Roberts and Wentzloff [25], which has a power consumption of 98 nW, but its wake-up range is at most 1.2 meters when the signal is transmitted at 0 dBm using patch antennas. Another receiver architecture with nano-power consumptions (specifically, 116 nW) has been presented by Oh et al. [26]. To avoid false awakenings their wake-up radio adopts two mechanisms: the first is to support addressing through the use of a 32-bit OOK-modulated CDMA code; the second is an automatic threshold feedback that detect the presence of interference sources and dynamically adapt the receiver's sensitivity. The authors tested the radio in three different frequency bands: the 402-405 MHz MICS band (Medical Implant Communication Service), the two ISM bands 915 MHz and 2.4 GHz. In these bands they measured the following sensitivities: -45 dBm (403 MHz), -43 dBm (915 MHz) and -41 dBm (2.4 GHz). They do not provide any actual wake-up range measurement. The bitrate of this radio is 12.5 kbps. Elgani et at. [27] presented an integrated circuit architecture for a wake-up receiver operating at 868 MHz with a 1 kbps bitrate. According to their simulations the integrated circuit can obtain the same sensitivity of WURs built with discrete components, i.e., -55 dBm, at the much lower power consumption of 13 nW.

A promising line of research on wake-up radio hardware envision the use of microelectromechanical systems (MEMS) for the realization of radio frequency components. In [34] Zhu et al, presented a resonant switch suitable to use in near-zero power RF wake-up receivers. This device showed a sensitivity of -8 dBm.

2.2 Wake-up radio-enabled protocols

Even though most of the research has been focused on new hardware receivers, both MAC, routing and cross-layer protocols designed to work with wake-up radios have been presented in the past years. Table 2.2 summarizes the networking protocols that we survey in this section.

One of the first MAC protocols that envisioned the use of wake-up mech-

Table 2.2: Wake-up radio-enabled networking protocols

Protocol	**Type**	**Implementation**
STEM [35]	MAC	Simulation
WUR-MAC [36]	MAC	Simulation
Stathopoulos et al. [37]	Routing	Testbed
ALBA-WUR [9]	Cross-layer	Simulation
FLOOD-WUP [22]	Routing	Simulation
GREEN-WUP [22]	Cross-layer	Simulation
CTP-WUR[38]	Routing	Simulation
GreenRoutes [39]	Cross-layer	Simulation
WHARP [40]	Cross-layer	Simulation
T-ROME [41]	Cross-layer	Testbed
Zippy [24]	Cross-layer	Testbed
OPWUM [42]	Cross-layer	Simulation

anisms for energy savings is STEM [35]. Two versions of STEM were proposed: STEM-T where the wake-up signal is just a tone and STEM-B where the wake-up signal contains the address of the destination node. Since at that time ultra-low power radios did not exist, the authors proposed to use two main radios, operating on different frequencies: one acting as a real main radio (i.e., transmitting data packets and being active only when needed) and the other one acting as a wake-up receiver. In order to have a low energy consumption by this second radio, it followed a very low duty-cycle. For this reason STEM does not solve the problems of known duty-cycling MAC protocols. WUR-MAC [36] is a simple MAC protocol that implements the well-known RTS/CTS handshake mechanism on wake-up radios. It requires that main and wake-up radio operate on the same frequency band, because the channel access packets are sent by the former and received by the latter one. It supports both unicast and broadcast communications, since the wake-up packet contains the destination address. It also supports channel selection, as sender and receiver use the RTS and CTS packets to agree on one of them.

The protocol presented by Stathopoulos et al. [37] considers a dual radio system where each node uses the low-power radio to request to a central node, called “topology controller” the end-to-end path to the destination. The controller, which is aware of the main radio paths between all the pair of nodes, uses the low-power radio to send back the path to the sender node and to wake-up all the nodes along the path. The wake-up addressing is ID-based. The data packets forwarding is handled by the high bandwidth radio.

Spenza et al. in [9] presented ALBA-WUR, a cross-layer protocol for data collection in networked sensing systems. The authors redesigned ALBA-R [43] to take advantage of the features of wake-up radios, and in particular of semantic addressing. This technique allows to represent the complex relay selection policy of ALBA-R in a pool of wake-up addresses which is dynamically updated by the nodes, depending on their status. In ALBA-R the next-hop selection is based on some parameters: the possible advancement towards the sink (represented by the Geographic Priority Index, GPI), whether the neighbor is congested and how good it forwarded past packets (represented by the Queue Priority Index, QPI) and whether the neighbor is on a path towards a dead end (represented by a color). In ALBA-WUR every node creates a pool of wake-up addresses that represent all the possible combinations made of the QPI values it can satisfy, all the possible number of packets it can hold in its buffer and all the possible colors that nodes must have in order to forward packets to it. In this way a node wakes-up only those neighbors that satisfy the forwarding policy, i.e. only those whose current status makes them the best relay. Petrioli et al. in [22] presented two protocols that exploit semantic addressing to improve system performance: FLOOD-WUP and GREEN-WUP. FLOOD-WUP is an energy efficient flooding protocol where nodes dynamically change their wake-up address to avoid being woken up by the retransmissions of packets they have already received. GREEN-WUP is a converge-casting protocol where the wake-up address of each node reflects its hop count from the network sink and its energy status (i.e., the remaining energy in the battery and the amount harvested from the environment, if any). The protocol chooses the next-hop with an RTS/CTS exchange, where the RTS is actually the wake-up request. The transmitting node sends a wake-up request containing the desired hop count (i.e., its distance from the sink - 1) and the desired energy level. If any neighbor satisfies this condition, it has the corresponding wake-up address, therefore it wakes up and replies with the CTS. Then the sender node selects the next-hop among all the neighbors that replied. Basagni et al. in [38] redesigned the Collection Tree Protocol (CTP), the *de* *facto* standard in WSN data collection routing, to make it work on wake-up radio-enabled wireless sensor networks. This protocol, called CTP-WUR, tackles the problem of the shorter range of WURs compared to the main radios. To virtually extend the wake-up range it enables relaying of wake-up sequences. In this way the node that will be woken up is not the parent in the routing tree, but the parent of the parent, which is assumed to be inside the range of the main radio. Thanks to this, CTP-WUR is able to reduce the number of hops needed to reach the sink and end-to-end latencies. Basagni et al. in [39] presented GreenRoutes, a cross-layer protocol for Wireless Sensor Networks with energy-harvesting and wake-up radio capabilities. In GreenRoutes a node chooses the next-hop through a RTS/CTS exchange. The selection is based on the distance from the sink and on the remaining energy along recently used routes to the sink. Thanks to semantic addressing only those neighbors that satisfy the requirements of the forwarding policy are woken up. In [40] Basagni et al. presented WHARP a cross-layer protocol for Wireless Sensor Networks with energy-harvesting and wake-up radio capabilities. In WHARP semantic addressing is used to wake-up neighbors that are closer to the sink: the wake-up address of each node is its distance in hops from the sink. When a node receives a wake-up request with a matching address, it decides whether to activate its radio and participate in the RTS/CTS exchange using a Markov Decision Process (MDP) based on the forecast energy and expected traffic load.

Kumberg et al. [41] presented a cross-layer data forwarding protocol: T-ROME. It uses RTS/CTS packets to reduce collisions over the wake-up messages. It is based on a tree routing algorithm: a node can send a wake-up packet only to its parent, up until it reaches the destination. A data packet, instead, can cross several levels of the routing tree, thanks to the main radio longer range, saving energy of the intermediate nodes. Two optimizations of T-ROME are proposed by Kumberg et al. in [44]: they allow significant savings in terms of energy consumptions and overhead. The first optimization avoids the transmission of MAC acknowledgements during the route establishment phase: this reduces the overhead of the protocol and the probability of a communication failure, because less packets are exchanged. The second optimization tackles the problem of the nodes that get disconnected from the network because their parents in the routing tree are temporarily unreachable or have run out of energy. In this case they can reconnect to the rest of the network if a parent of the disappeared node is within their wake-up range.

In [24] Sutton et al. introduced Zippy, an on-demand flooding protocol. Zippy can be seen as made of three phases. The first one is asynchronous network wake-up, where all the nodes in the multi-hop network are woken up. The second is neighborhood synchronization, that allows fine-grained per-hop synchronization in the order of few tens of microseconds. Then, there is bit-level data dissemination. To avoid destructive interference due to concurrent transmissions, carrier frequency randomization is used. The protocol has been validated in the multi-hop testbed FlockLab. Aoudia et al. [42] presented OPWUM, a cross-layer protocol exploiting wake-up radios to opportunistically select the next hop relay. It generalizes the concept of timer-based contention mechanism: in fact the authors do not specify any metrics according to which the choice should be made, leaving it to be defined depending on the specific application. The RTS/CTS exchange happens completely through the wake-up radio.

Chapter 3

Wake-up Radio Ranges: A Performance Study

To build a testbed for wake-up radio-based wireless sensor networking, it is necessary to know how far a wake-up signal can be heard by the wake-up receiver used. In this chapter the results of the performance evaluation of a wake-up radio receiver prototype are showed and discussed. The evaluation consists in several ranging experiments, both indoors and outdoors, where a transmitter sends a set of wake-up sequences to a receiver positioned meters away. Every received sequence is compared against the transmitted one to check whether it has incurred errors or not. In Section 3.1 the wake-up radio design and the wireless mote used are described. Section 3.2 details the experimental setup, while Section 3.3 presents the investigated performance metrics. In Section 3.4 and Section 3.5 the results of the experiments are presented.

3.1 Wake-Up Radio Receiver and mote architecture

3.1.1 Wake-up radio receiver

The architecture of the WuR used in this work is depicted in Figure 3.1 [9]. In this section we summarize its main components.

The receiver uses the On-Off Keying (OOK) modulation, one of the simplest forms of Amplitude-Shift Keying modulation. Digital data are

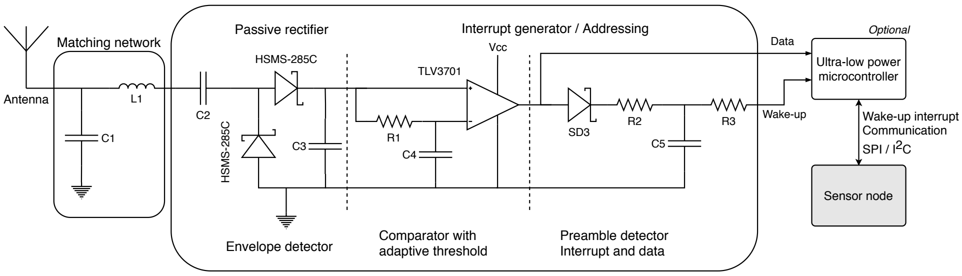


Figure 3.1: Architecture of the wake-up receiver [9].

represented as the presence or absence of a carrier wave.

The first component is the *matching network*: it maximizes the power transfer from the antenna to the rest of the circuit. It consists of an LC filter, whose components are set based on the transmission frequency. In this work, we tested two different prototypes of the WuR, optimized to work either in the 868 MHz frequency or in the 433 MHz frequency.

The output of the matching network is an RF signal from which information (i.e., the wake-up sequence) can be recovered. This task is carried out by the next two components. The *passive envelope detector* discards the frequency and phase content of the modulated waveform and only detects its amplitude, consistent with using OOK modulation. In the tested prototypes it consists of a single-stage half-wave rectifier with series diodes. In particular, the HSMS-285C diodes from Avago Technologies were used. They are optimized for sub-GHz frequencies and for incoming power levels lower than -20 dBm. They achieve a sensitivity of -57 dBm.

Once the signal is rectified, the bits of the received wake-up sequence are reconstructed by using an *ultra-low power comparator*. In order to reduce the static power consumption of the circuit, an adaptive threshold mechanism keeps the V^- pin of the comparator at half of the input signal level. This allows to avoid the use of a voltage divider, as the threshold is generated using the energy coming from the antenna. The choice of the comparator influences the power consumption of the WuR, because it is the only active component of the design, as well as its overall sensitivity. In fact, comparators with a lower voltage offset are able to sense smaller signals. However, such comparators typically have higher current consumption. The considered prototypes feature the TLV3701 comparator from Texas Instruments, which has a reasonable current consumption of 560 nA.

The last component of the WuR is the *preamble detector*. It is used to filter out interference due to noise or other communications that can trigger unwanted awakenings. For this purpose, a preamble is added at the beginning of the wake-up sequence. The preamble is an OOK modulated signal representing a specific sequence 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 discarded by the RC section of the preamble detector, which works as a low-pass filter. If a preamble is received at the f_p bit rate, the preamble detector generates a wake-up interrupt to either the sensor node's MCU or to an optional on-board ultra-low power micro-controller that reads data directly from the output of the comparator and performs address matching.

The optional ultra-low power micro-controller (e.g., the PIC12LF1552 from Microchip [45]) relieves the sensor node of the wake-up address recognition task and allows it to remain in the sleep state until a valid address is received. Hence, it decreases the total energy consumption.

This design obtains desirable characteristics for a WuR, namely, very low power consumption ($< 1.3\ \mu$W) and high sensitivity (up to -55 dBm).

3.1.2 Mote architecture

In our experiments we used the MagoNode++ [31], a wireless mote designed for WSN applications, which features the wake-up radio receiver described in Section 3.1.1. Figure 3.2 shows a snapshot of a MagoNode++ featuring the WuR prototype on top of the casing used in our experiments.

The MagoNode++ is a low-power wireless mote equipped with a micro-controller/transceiver bundle. The microcontroller is based on a 8-bit, 16 MHz CPU featuring 256 KB of ROM and 32 KB of RAM. The integrated transceiver is a 2.4 GHz, 802.15.4 compliant radio module connected to a radio front-end that gives higher radio performance with low-power consumption. The wake-up extension board designed for the MagoNode++ is made up of two main components: The wake-up transmitter and the WuR. As mentioned, these components differ depending on the wake-up frequency used, namely, 868 MHz or 433 MHz. The wake-up transmitter is based on the CC1101 transceiver from Texas Instruments, a low-power sub-1 GHz transceiver that supports OOK modulation [10]. The wake-up receiver is the one described in Section 3.1.1. Wake-up transmitter and receiver use two separate antennas with a 50 Ω impedance and gain depending on the

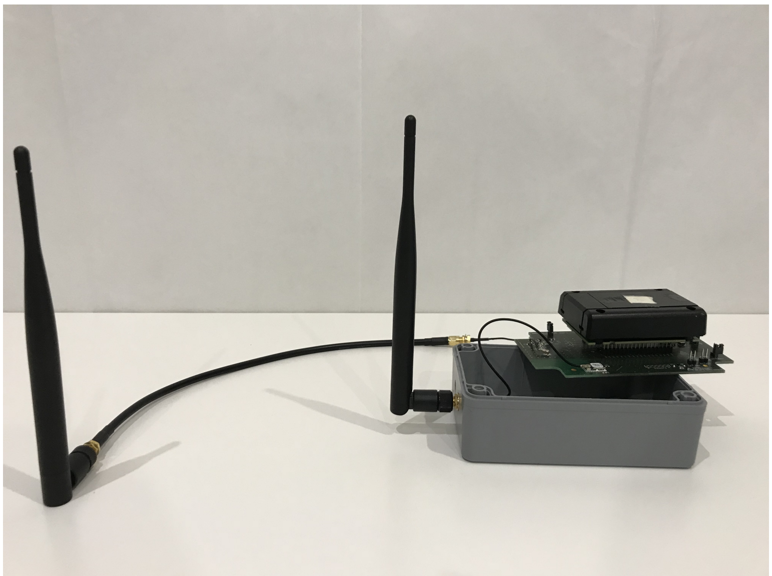


Figure 3.2: A MagoNode++ mote with a WuR.

frequency. The two wake-up antennas are connected to the wake-up transmitter and to the WuR circuitry. In order to avoid any distortion of the emitted electromagnetic fields, in our experiments we keep these two antennas at a distance δ higher than the near-field distance, which depends on the frequency and on the size of the antenna. The impedance matching of the RF section was carefully designed to provide the maximum power transfer between the antenna and the rest of the circuit. Details on the tuning of the matching network have been presented in [46]. The designers also chose to use separate antennas for the wake-up system and the main radio (which is integrated in the MagoNode board) for greater accuracy in assessing the performance of the WuR. Wake-up signals generated by the WuR are sent to the MagoNode++ through a low-level asynchronous interrupt pin of the microcontroller. Asynchronous interrupts are able to wake-up the MagoNode++ from the dormant state, in contrast with synchronous interrupts that can be triggered only when the node is in idle mode.

3.2 Scenarios and settings

In our experiments we used the MagoNode++ mote with WuR capabilities described in Section 3.1. We tested WuR prototypes operating at 868 MHz and at 433 MHz.

We measure WuR range performance by placing a sender node S and a receiver node R at a given distance d, with d varying in $\{3, 6, 9, \ldots, 24\}$ meters. We consider both indoor and outdoor settings, all located within the premises of the Computer Science department of the University of Rome "La Sapienza," Italy. Our indoor experiments were performed in a corridor of the building (Figure 3.3). The outdoor space is the building courtyard.

Figure 3.3: Indoor setting.

Sender and receiver exchange wake-up sequences each 8 bits long. Because of hardware constraints, the sequences contain no six 1 in a row. This results in 248 *viable sequences* that can be received by the WuR.

Each sequence is preceded by a 4-bit preamble to allow the WuR to signal the wake-up interrupt needed to start the address matching phase. The transmission power of the wake-up sequences is set to 10 dBm.

3.3 Investigated metrics

We selected 50 among all viable sequences. Each selected sequence s is set as the wake-up sequence of node R. Node S then sends sequence s to node R. This is repeated 25 times in total, with each copy being sent every 100 ms.

For each sequence, we measure the following:

1. *Sequences delivered*: The percentage of sequences received by node R.
2. *True positives*: The percentage of sequences received correctly (node R would correctly wake-up).
3. *False negatives*: The percentage of sequences received incorrectly (node R would not wake-up while it should).

In order to evaluate the likelihood that a transmitted wake-up sequence is modified into another valid one by the channel interference, causing an unwanted wake-up (i.e., a false positive), we do the following. Each sequence s is transformed into a sequence s' by flipping one of its bits, chosen randomly. If s' is not a viable sequence (i.e., it contains six 1 in a row) it is discarded, and a new sequence s' is created. If s' is viable, node S sends sequence s' to node R. This is repeated 25 times, with one transmission every 100 ms. For each flipped sequence, we measure the following:

1. *Flipped sequences delivered*: The percentage of sequences s' received by node R.
2. *True negatives*: The percentage of sequences s' that have been received different from sequence s (node R would correctly stay asleep).
3. *False positives*: The percentage of sequences s' that have been received incorrectly as sequence s (node R would wake-up while it should not).

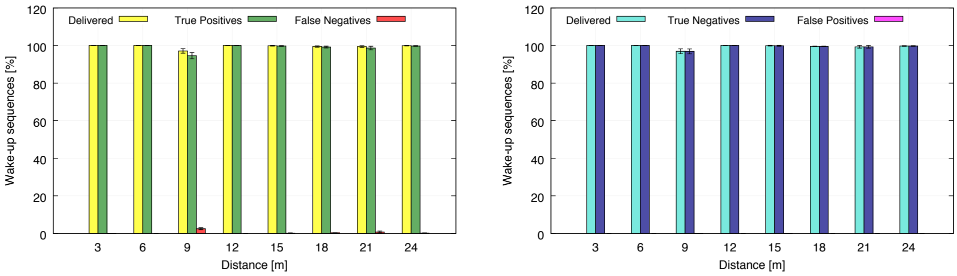
The results show the average of the outcomes of 5 repetitions of the experiment at each distance d.

3.4 Performance results

3.4.1 868 MHz

Results of indoor experiments are shown in Figure 3.4a and Figure 3.4b for viable and flipped sequences, respectively.

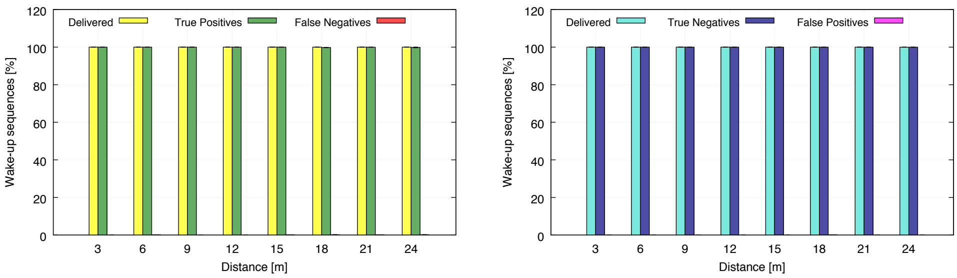
The percentage of delivered sequences is above 96% over all the distances. False Negatives are consistently under 1% except when the distance between sender and receiver is 9 meters. In this case 2.5% of the sequences incur errors. False Positives are negligible.



(a) Viable sequences (b) Flipped sequences

Figure 3.4: Indoor scenario, 868 MHz.

Results of outdoor experiments are shown in Figure 3.5a and Figure 3.5b for viable and flipped sequences, respectively.



(a) Viable sequences (b) Flipped sequences

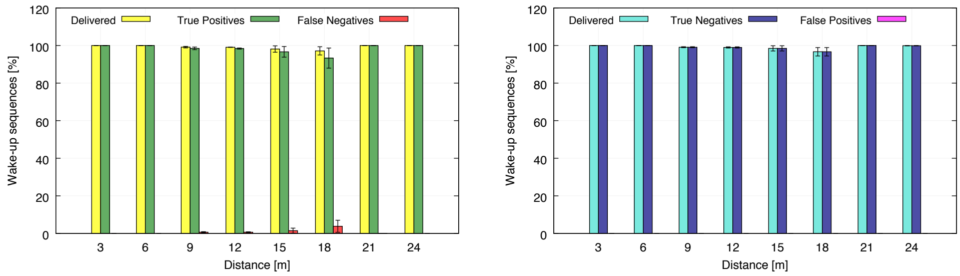
Figure 3.5: Outdoor scenario, 868 MHz.

At least 99.97% of the sent sequences were delivered to the receiver. These sequences were almost always correctly detected, since False Negatives are consistently under 0.2%. The only False Positives have been observed at 18 meters in an insignificant amount (0.02%).

3.4.2 433 MHz

Results of indoor experiments are shown in Figure 3.6a and Figure 3.6b for viable and flipped sequences, respectively.

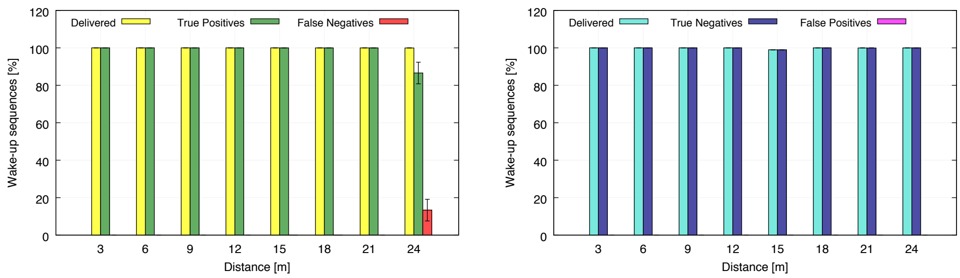
The percentage of delivered sequences is above 96% over all the distances. At 15 and 18 meters there is a small amount of False Negatives, i.e., of viable sequences received with some errors: 1.48% and 3.84%, respectively. At the same distances there is also a significant increase in the variance of the results.



(a) Viable sequences (b) Flipped sequences

Figure 3.6: Indoor scenario, 433 MHz.

Results of the outdoor experiments are shown in Figure 3.7a and Figure 3.7b for viable and flipped sequences, respectively. At least 99.92% of the



(a) Viable sequences (b) Flipped sequences

Figure 3.7: Outdoor scenario, 433 MHz.

transmitted sequences were delivered to the receiver. The viable sequences were almost always correctly detected up to 21 meters (a negligible amount of False Negatives has been observed at 15 and 18 meters). At 24 meters the performance begins to degrade, as 13% of the sequences incurs errors. During both indoor and outdoor experiments no False Positives have been received.

In general, for practical purposes both prototypes work well up to 21 meters. This provides us with a distance that can be safely used in larger testbeds and in simulation models. The slightly worse results in the indoor scenario are to be ascribed to the environmental noise.

3.5 Extended evaluation

To further evaluate the wake-up range performance of the MagoNode++ mote, we performed more experiments. This set of experiments gives more insights on how the WuR performs in scenarios with different levels of noise and interference and whether it is affected by the relative position between sender and receiver. We also evaluated how Hamming-encoding the wake-up sequences affects the wake-up ranges. The experiment setup is the one described in Section 3.2, but with two differences:

- We used two different indoor environments, one with a high level of noise and another with a low level of noise
- We tested the 868 MHz prototype.

We used the same 50 sequences of the previous experiments. Each selected sequence s is set as the wake-up sequence of node R. Node S then sends sequence s to node R. This is repeated 25 times in total, with each copy being sent every 100 ms. For each sequence, we measure the following metrics:

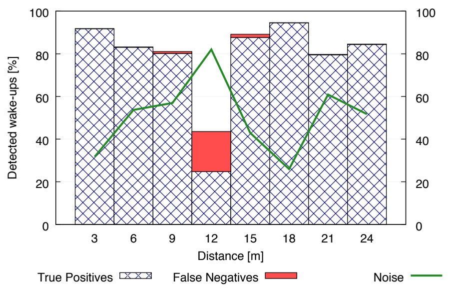
- *True positives*: The percentage of sequences received correctly (node R would wake-up)
- *False negatives*: The percentage of sequences received incorrectly (node R would not wake-up while it should)
- *Spurious messages*: The percentage of noise wake-up sequences received over the total number of received sequences.

The results are averaged over 5 repetitions at each distance d.

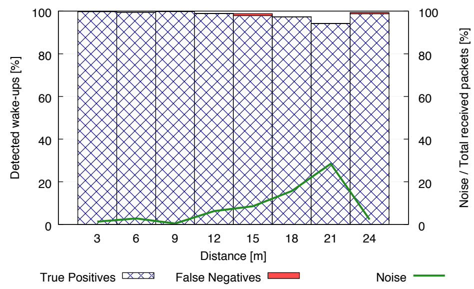
The high noise scenario is the same indoor corridor of the experiments of Section 3.4: it is located at the last floor of the building, therefore it is exposed to external sources of interference. The low noise scenario is another corridor in the same building, but located at a lower floor, therefore more protected by the surrounding buildings. The outdoor scenario is the same: the courtyard of the building.

3.5.1 Range tests without Hamming encoding

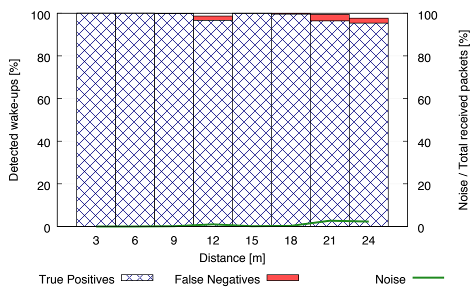
Figure 3.8a, Figure 3.8b and Figure 3.8c show, respectively, the results of the indoor tests in a high noise scenario and in a low noise scenario and the results in an outdoor scenario, when the wake-up sequence is not Hamming-encoded (as in Section 3.4). In this case the length of the transmitted messages is 2 bytes: the first contains the preamble and the second the actual wake-up sequence.



(a) High noise indoor scenario



(b) Low noise indoor scenario



(c) Outdoor scenario

Figure 3.8: Results of range experiments - Without Hamming encoding

In the same indoor scenario of the previous experiments the percentage of transmitted wake-up sequences that have been received is above 90% only two times, at 3 and 18 meters. At 12 meters the worst performance of the wake-up receiver has been recorded: 43.5% of the transmitted sequences being received and 24.8% of True Positives. The percentage of noise messages over the total number of received wake-ups goes from 26% at 18 meters to 82% at 12 meters. On average, over all the distances, 50.7% of the received wake-up sequences was noise. As we can see from the figures the percentage of detected sequences is inversely proportional to the noise messages.

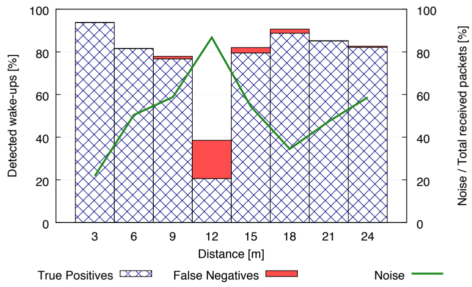
In the low noise scenario more than 94% of the transmitted sequences were delivered to the receiving node across all the distances. True positives are consistently above 97% with a slight decrease to 94.2% when the distance between the two nodes is 21 meters. This corresponds to the maximum value of the percentage of spurious messages received (28.4%).

In the outdoor scenario the percentage of received wake-up sequences is above 97% across all the distances. True Positives are consistently above 99%, with the exception of 12, 21 and 24 meters, where False Negatives amount to, respectively, 2%, 3% and 2.3%. Spurious messages are below 3%.

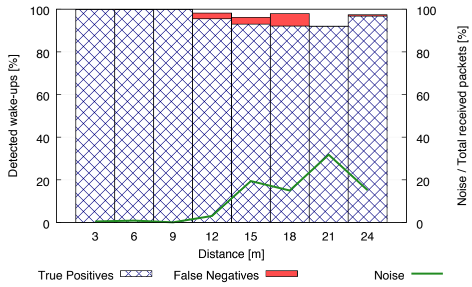
The lower amount of received sequences in the high-noise scenario is explained by the higher percentage of spurious messages received. In fact, if the wake-up receiver is triggered during the interval between two consecutive transmissions of a wake-up sequence (set at 100 ms), it may not be able to start receiving a *real* wake-up sequence. In general, in both scenarios, the presence of noise affects the ability to receive at all a wake-up sequence, while it does not have a big impact on the False Negatives, that are, in general, below 1%. Instead, at 12 meters in the high noise scenario, False Negatives are 18.8%, while the spurious sequences are 82% of the total amount of received sequences. We noticed that 12 meters in the high-noise corridor correspond to the only position where the receiving MagoNode++ has a window on both sides, while at all the other positions there is at least one wall. Since the corridor is located at the last floor of the building, that position is the most exposed to external interference. The most probable source of interference at 868 MHz is some 4G-LTE base station operating on the 800 MHz band (from 791 MHz to 862 MHz). This explains the different level of noise between the last floor of the building and the lower floor and the courtyard, that are more protected by the surrounding buildings.

3.5.2 Range tests with Hamming encoding

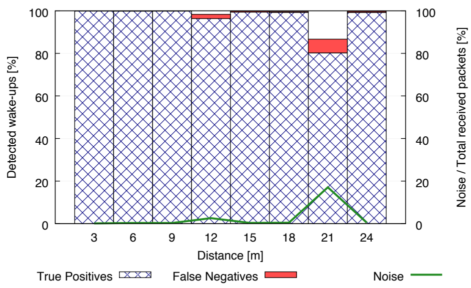
Figure 3.9a, Figure 3.9b and Figure 3.9c show, respectively, the results of the indoor tests in a high noise scenario and in a low noise scenario and the results in an outdoor scenario, when the wake-up sequence is Hamming-encoded. The total length of the transmitted messages is 4 bytes. The first byte is the preamble. The second byte contains *sync* bits. The actual encoded wake-up sequence occupies the last two bytes. In fact, the $H(8,4)$ Hamming code used by the MagoNode++ takes 4 bits of data and produces a 8 bit codeword. Hence, the 8-bit wake-up sequence becomes 16 bits long.



(a) High noise indoor scenario



(b) Low noise indoor scenario



(c) Outdoor scenario

Figure 3.9: Results of range experiments - With Hamming encoding

In both scenarios higher values of spurious wake-ups have been received compared to the experiments without Hamming encoding discussed in Section 3.5.1. This affects the results, especially in the low-noise scenario, where slightly higher levels of False Negatives have been received, with respect to the results of the experiments with no Hamming encoding.

In the low noise scenario more than 92% of the transmitted sequences were delivered to the receiving node across all the distances. True positives are above 95% from 3 to 12m and at 24m. Between 15 and 21m, in correspondence of higher values of noise (between 15% and 32% of the total number of received sequences), True Positives are between 92% and 93%.

Also in this case, in the high noise scenario the percentage of received wake-up sequences is above 90% only at 3 and 18m, while at 12m only 38% of the transmitted sequences were received, due to the high percentage of spurious sequences (87%). Only at 3m more than 90% of the transmitted wake-up sequences were correctly received.

In the outdoor scenario the percentage of received sequences is above

99% across all the distances, with the exceptions of 12 and 21 meters. The percentage of received sequences almost corresponds to the percentage of True Positives, because wrong sequences are always less than 0.5%. At 21 meters the True Positives fall down to 80%, with a percentage of False Negatives of 6.4%. This happened in correspondence with a "spike" of noise messages, that totaled the 17% of the total amount of received sequences.

We can conclude that, at least in these simple experiments where just one node sends wake-up sequences to another one, the Hamming encoding does not give any remarkable advantage that can justify the longer transmission time. In the high-noise scenario the levels of interference may be too high for a code that can correct only one wrong bit to be effective.

3.5.3 Range tests with receivers around the transmitter

With these experiments we wanted to understand whether the relative position (i.e., the angle) between the transmitter and the receiver has some effect on the percentage of wake-up sequences received correctly. We placed 8 different MagoNode++ motes around the transmitter in a wide open area in the Villa Borghese gardens in Rome, Italy. Figure 3.10 shows how the nodes were placed; the arrows show the side of the antennas on the motes. The distance d between sender and receivers varies in $\{3, 6, 9, \ldots, 21\}$ meters.

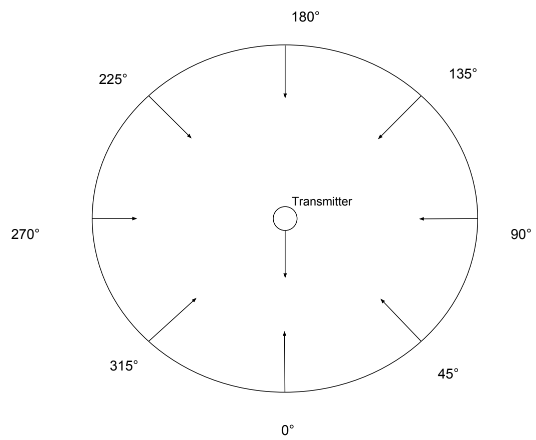
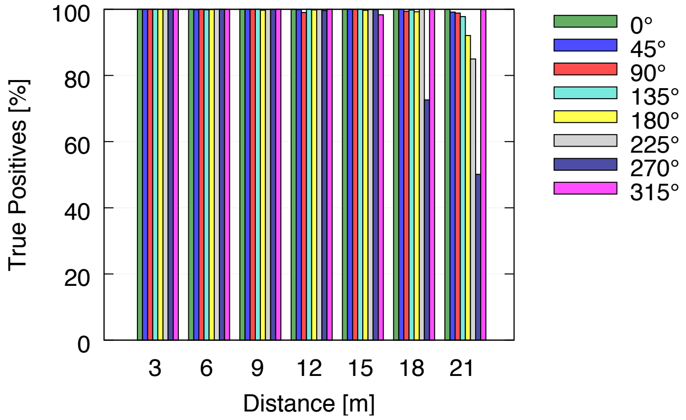
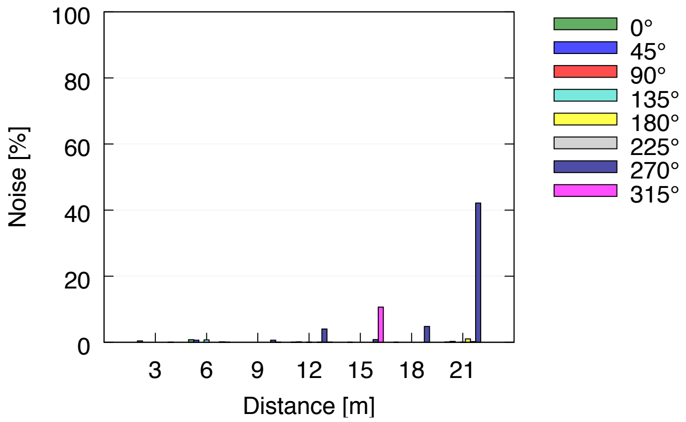


Figure 3.10: Positions of the nodes during the experiments

Figure 3.11a and Figure 3.11b show the outcome of the experiments, respectively in terms of True Positives and Noise wake-ups, when no Hamming encoding was used. Up until 15 meters the percentage of True positives is consistently above 98% from all the angles around the sender.



(a) True Positives

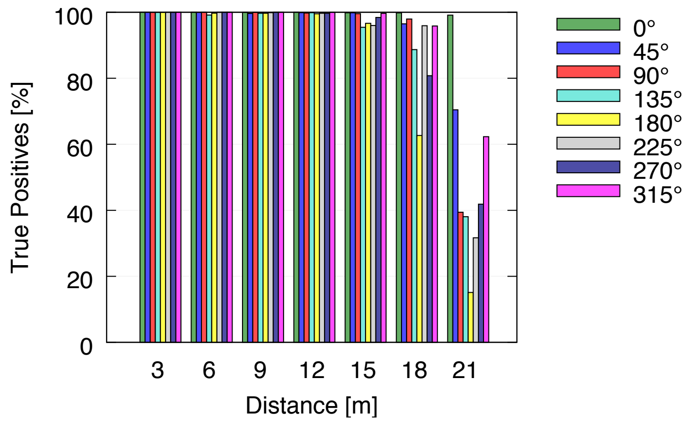


(b) Noise

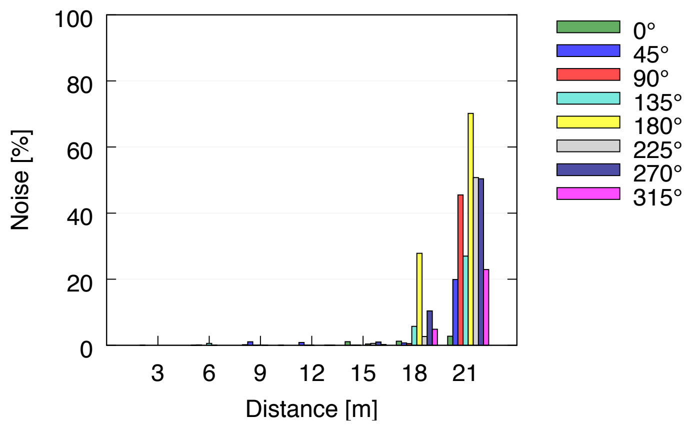
Figure 3.11: Receivers around the transmitter without Hamming encoding.

At 18 meters the percentage of True Positives stays on the same level at all the angles with the exception of 270°, where it falls at 72.6%. At 21 meters the percentage of True Positives at 270° falls to 50%. Also at 180° and 225° the percentage of correctly received wake-up sequences starts to decrease: 92% and 85%, respectively. At all the other angles it remains above 98%. During all these experiments interference has almost never been a real concern. Only the node placed at 270° at 21 meters seemed to be affected by some noise (42% of the total amount of received sequences).

Figure 3.12a and Figure 3.12b show the results of the experiments, respectively in terms of True Positives and Noise wake-ups, when the wake-up sequence is Hamming-encoded.



(a) True Positives



(b) Noise

Figure 3.12: Receivers around the transmitter using Hamming encoding.

Until 12 meters the percentage of True positives is above 99% for all the angles around the sender. At 15 meters it is still above 95% for all the angles.

At 18 meters, some positions show remarkable reductions of the amount of correctly received wake-up sequences. In particular, behind the transmitter (180°) only 63% of the sequences were correctly received. At 135° and 270° the percentage of True Positives was, respectively, 88.7% and 81%. At 21 meters the percentage of correctly received wake-up sequences had its maximum value, 99%, in front of the transmitting mote (0°) and its minimum value, 15%, behind the transmitter (180°). At the other positions the percentage of True Positives decreases from the angles in front of the transmitter (45° and 315°, the other positions where the True Positives were above 50%) to the angles behind it.

We can notice, just like the experiments in Section 3.5.1 and 3.5.2, that in correspondence with very low values of True Positives the receiving motes recorded high percentages of spurious wake-ups (i.e., from 20% to 50% of the total number of received sequences).

3.6 Conclusions

We have evaluated the range performance of a wake-up radio receiver designed to have high sensitivity and very low power consumption. We tested two versions of the WuR optimized to work at different two frequencies: 868 MHz and 433 MHz. We sent a pool of wake-up sequences between two motes placed at increasing distances and determined how many of them were received and whether they were received with errors or not. We then intentionally flipped one bit of every sequence in the pool and repeated the experiment to see if potential interference could alter the transmitted sequence and produce a False Positive. The experiments were run at both indoor and outdoor locations. We found that the tested WuR can receive more than 96% of the transmitted sequences, on both frequencies, indoors. On the 868 MHz frequency we observed that less sequences are received with errors. Both the receivers perform better outdoors, with more than 99% of the sequences being received at distances up to 21 meters. False Positives were rare across all experiments.

We further evaluated the 868 MHz wake-up receiver in three scenarios with different levels of noise. We also investigated whether the relative position between transmitter and receiver has some effect on the wake-up sequence reception. Our results showed that in a noisy environment less than 90% of the transmitted sequences can be received, irrespective of whether the wake-up message is encoded with a Hamming code.

Chapter 4

Wake-up radio-based WSNs: A networking performance evaluation via real-world testbed experiments

In this Chapter we show the benefits of wake-up radio technology through a set of experiments performed on a real-world testbed.

We compare the Collection Tree Protocol (CTP) [47] with duty cycle against CTP-WUR [38], an extension of CTP designed to exploit wake-up radio receivers. We implemented CTP-WUR in TinyOS, extending the implementation of CTP distributed with TinyOS 2.1.

We also present an optimization of CTP-WUR based on the insights obtained from the testbed experiments. The in-field evaluation of this variant confirms improvements on both end-to-end latencies and energy consumption.

We refer to CTP with duty cycle as CTP-LPL because duty-cycling is obtained through Low Power Listening (LPL).

4.1 CTP

The Collection Tree Protocol (CTP) [47] is the *de facto* routing strategy when it comes to data collection in Wireless Sensor Networks, since it is widely used in several deployments and testbeds. It is a distance vector protocol that builds and maintains a minimum-cost tree rooted at the network sink. Each node forwards data packets to its parent node in order to move them toward the sink.

Each node in the network estimates the cost of its route to the sink in terms of Expected Transmissions (ETX), i.e., the average number of transmissions needed to send a packet from a node to the sink. This cost is given by the sum of the cost advertised by the parent node plus the estimation of the cost of the direct link between the node and its parent. This estimation considers the quality of both the ingoing and outgoing links. The outgoing link quality is given by the ratio between the number of data packets sent (including retransmissions) and the number of received acknowledgements. The ingoing link quality is given by the ratio of the number of beacons received from the parent node over the total number of beacons sent by the parent [48]. The ETX of the sink is zero.

To build and maintain the routing tree the nodes exchange their local cost estimates by broadcasting control beacons. The beacon transmission rate is not constant but it is regulated by a mechanism called *adaptive beaconing*, based on the Trickle algorithm [49]. When a node has to send a beacon it waits for a random time within the interval $[\tau/2, \tau]$. After each beacon transmission the value of τ is doubled, until it reaches the maximum value τ_{max}. A node reset τ to the minimum value τ_{min} when a better route to the sink or when a routing problem (e.g., a loop) are discovered. In this way CTP reduces the beacon transmission frequency (and its energy consumption) as the network topology gets more stable, but it is still able to quickly react to the changes of the radio links. CTP is also able to early detect possible inconsistencies or loops in the topology using a mechanism called *datapath validation*. Each data packet contains the ETX of the transmitter. The receiver compares this value against its own route cost: if the receiver's ETX is not smaller than the transmitter's, routing information is assumed to be stale and the receiver starts a topology repair by resetting the beacon transmission frequency (i.e., setting τ back to τ_{min}).

Figure 4.1 shows an example of how CTP works when the nodes use LPL to reduce their consumption. They periodically activate the main radio. When a node wants to send a packet it starts sending a long preamble to intercept the next active period of its parent node, which will keep its radio

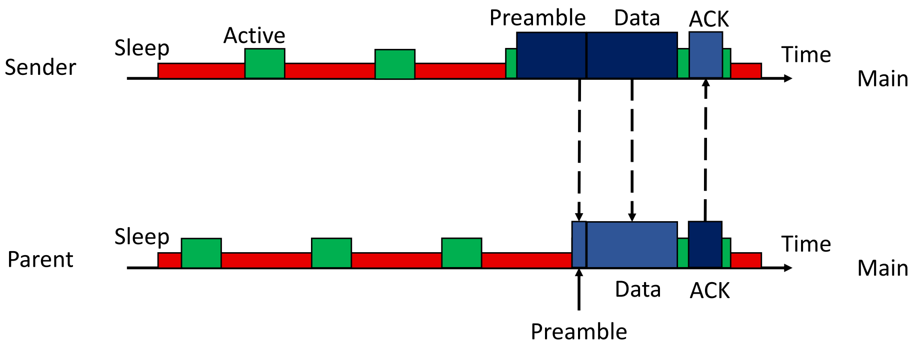


Figure 4.1: Example of data forwarding using CTP-LPL

on, waiting for the following packet transmission. The parent sends back an ACK to the sender and they both switch off their radios, continuing to follow their duty-cycles.

4.2 CTP-WUR

CTP-WUR [38] is an extension of CTP for wake-up radio-enabled Wireless Sensor Networks. It is designed to mitigate the fundamental limitation of this kind of networks, i.e., routes are significantly longer than those of networks that use only the main radio. It is based on the assumption that a node, thanks to the higher transmission range of its main radio, can directly communicate with a node that is further away than its parent node. To allow this direct communication, in CTP-WUR wake-up sequences are relayed through intermediate nodes in order to activate the distant node. The relaying is performed by the intermediate nodes through their wake-up radios. CTP-WUR achieves energy savings as the intermediate nodes do not to use their main radios; it also reduces latencies as data packets virtually go through more network hops with less transmissions. In CTP-WUR each node has three different wake-up addresses: a unicast wake-up (WUR) address made of its unique network identifier, a WUR relay address that consists of its network identifier and an additional flag to indicate that the wake-up request should be forwarded and a broadcast wake-up address shared by all the nodes in the network.

The operation of CTP remains unchanged, but every packet transmission is preceded by a wake-up phase in which the sender node wakes up the

Algorithm 1: Beacon Transmission

$\tau \leftarrow \tau_{min}$
while *True* **do**
    $waitTime \leftarrow random(\tau/2, \tau)$
    wait($waitTime$)
    updateRoutingInfo( )
    $beaconPkt \leftarrow newBeacon()$
    sendWUR(wurBroadcastAddress)
    wait(wurBroadcastTime)
    TurnOnRadio( )
    send(broadcastAddress, $beaconPkt$)
    TurnOffRadio( )
    if $\tau < \tau_{max}$ **then**
        $\tau \leftarrow 2\tau$
end while

intended receivers. When a beacon has to be transmitted, the sender awakes all its neighbors by sending a wake-up sequence containing the broadcast wake-up address. Nodes receiving this message activate their main radios and await the actual beacon transmission. After transmitting the wake-up sequence, the sender turns on its main radio and, before sending the beacon, awaits for a given amount of time to allow the activation of the receivers. After the beacon transmission the sender goes back to sleep, as its neighbors do after they received the packet.

When a node has to send a data packet, it tries to forward it directly to the parent of its parent node in the routing tree (i.e., its grandparent). Figure 4.2 shows an example of data forwarding in CTP-WUR. When a node has a data packet to send, it transmits to its parent a wake-up request containing the WUR relay address. It also starts a timer to give enough time for the relaying of the wake-up packet and the activation of the grandparent's main radio. The parent node sends a wake-up request to its own parent, i.e., the grandparent of the sender node, using its unicast WUR address. When the grandparent receives the wake-up sequence, it activates its main radio and awaits for the packet transmission from the sender. The sender transmits the data packet to its grandparent and awaits for the acknowledgment before turning off the main radio and going back to sleep. When the grandparent receives the data packet, it sends back an ACK and turn off its main radio.

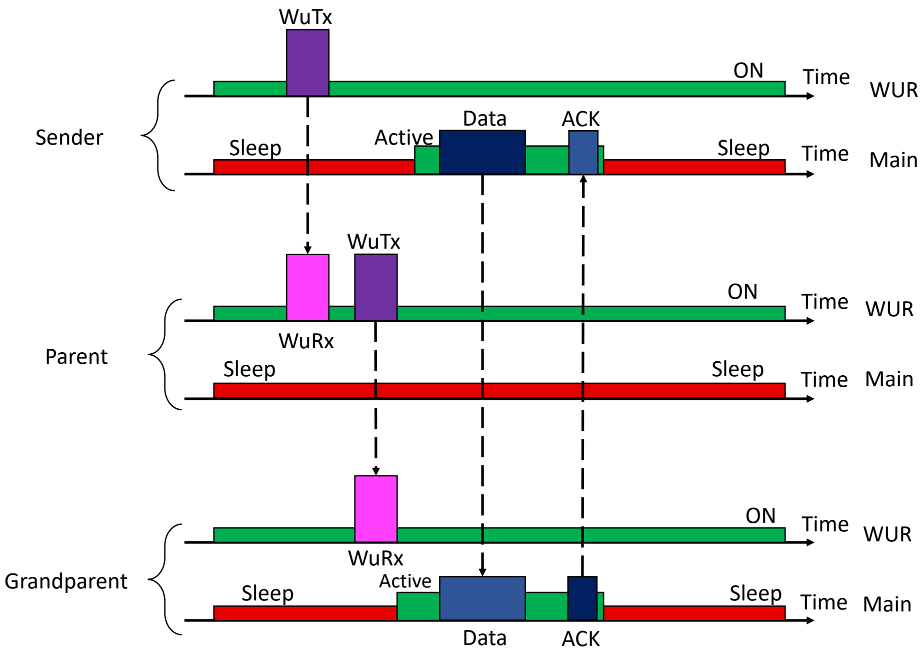


Figure 4.2: Example of data forwarding using CTP-WUR

It may happen that data packets are not received by the grandparent, as it may be out of range or temporarily unreachable due to variation of channel conditions. For these reasons, the direct transmission to the grandparent is repeated for a maximum number of times T. If all the attempts fail, the sender tries to forward the packet to its parent, for another T times, before dropping it. To do this the sender will transmit a wake-up request containing the parent's unicast WUR address.

4.2.1 Improving CTP-WUR

The assumption that the grandparent node lays within the main radio range of the sender node may not always be true. In fact, due to the variability of the radio channel conditions, the two nodes may be, even just temporarily, out of range. If this condition persists for several data packets all the attempts to skip the intermediate hop actually lead to a waste of energy and to increased packet latencies. For this reason, we propose a variant of CTP-WUR that is able to recognize when a node is spending too much time trying to forward packets to its grandparent, with no success.

Algorithm 2: Data Transmission

```
Receive packet, dataPkt, from upper layer
retries ← 2T
while retries > 0 do
    if retries > T then
        sendWUR(parentRelayAddress)
        wait(wurFwdTime)
        TurnOnRadio( )
        send(grandParent, dataPkt)
        wait(ACKTimer)
        TurnOffRadio( )
        if dataPkt is acknowledged then
            return
        retries ← retries − 1
    else
        sendWUR(parentAddress)
        wait(wurTime)
        TurnOnRadio( )
        send(parent, dataPkt)
        wait(ACKTimer)
        TurnOffRadio( )
        if dataPkt is acknowledged then
            return
        retries ← retries − 1
end while
if retries == 0 then
    Drop packet
```

Algorithm 3: Reception of a wake-up packet

```
address ← received WUR address
if address == WurBroadcastAddress then
    TurnOnRadio( )
    wait(packetRxTime)
    TurnOffRadio( )
    send received beacon to upper layer
else if address == WurRelayAddress then
    sendWUR(parentAddress)
else if address == WurAddress then
    TurnOnRadio( )
    wait(packetRxTime)
    TurnOffRadio( )
    send received data packet to upper layer
```

Each node keeps track of who acknowledged the last w data packets it has forwarded: either the grandparent or the parent node. It then computes the percentage of packets acknowledged by the grandparent. If this percentage is below a given threshold th, it means that over the period of time during which the node has forwarded the last w packets it had not been able to consistently deliver them to the grandparent and it had to fall back on the transmission to its parent. In this case, in order to avoid further waste of energy and time, the sender node will temporarily stop trying to forward the next data packets to its grandparent and will try to send them to its parent node. To do this the sender will transmit a wake-up request containing the parent's unicast WUR address. The normal CTP-WUR behavior is restored (i.e., the node tries to forward packets to the grandparent in the first place) when one of the following conditions occurs:

- the topology changes (the node has a new parent node, or a new grandparent).
- the main radio link quality of the node with its parent has improved.
- the main radio link quality between the parent node and its parent improved.

If the percentage of packets acknowledged by the grandparent is greater or equal to the threshold th the sender keeps forwarding packets to the grandparent. This percentage is computed every w packets: after every time it is computed, the counters for the number of the handled packets and the number of packets acknowledged by the grandparent is reset. Changing the value of w allows to modify the size of the "sampling window", making CTP-WUR faster or slower to recognize the possible lack of connectivity with the grandparent.

If we indicate with s a node that has to transmit a data packet, with p its parent node in the routing tree built by CTP-WUR and with gp the parent of its parent node, the decision of whether s will ask p to wake-up gp on its behalf can be shown in mathematical notation as:

$$nextHop(s) = \begin{cases} gp & \text{if } \frac{ACK_{GP}}{SENT} \geq th \\ p & \text{otherwise} \end{cases} \tag{4.1}$$

where ACK_{GP} is the number of packets successfully acknowledged by the grandparent and $SENT$ is the total number of packets sent.

4.3 Experimental scenario and settings

We deployed 11 MagoNode++ motes outdoors, in Villa Borghese gardens in Rome (Figure 4.3). They are equipped with the wake-up radio operating at the 868 MHz frequency. Each node is powered by a single Li-ion battery with a nominal voltage of 3.7 V and a nominal capacity of 800 mAh.

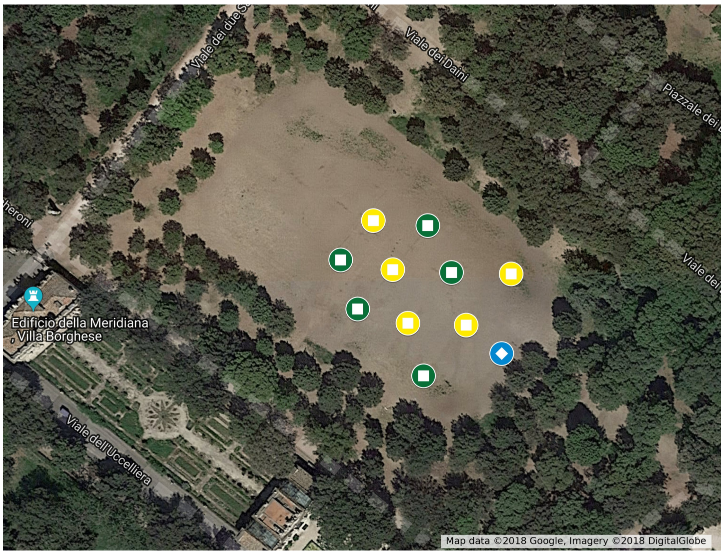


Figure 4.3: Satellite view of the testbed: The blue marker is the sink; the source nodes are shown in green while the rest of the nodes in yellow.

The distances between the nodes are shown in Figure 4.4 and are based on the results of the wake-up range evaluation discussed in Chapter 3. However, such short distances are easily covered by the main radio of the MagoNode++. Therefore, in order to have a multi-hop scenario we set the transmission power of the main radio to its minimum value, -2 dBm, for both CTP-WUR and CTP-LPL experiments.

Of the 11 nodes of the network, 5 of them are *source* nodes, i.e., they generate data packets (shown in green in Figure 4.3) and forward those received by the other nodes; another 5 nodes act only as relays (shown in yellow in Figure 4.3), while the remaining node is the sink of the network (showed in blue in Figure 4.3). Each source node generates data packets at a rate of 1 pkt/min. In order to avoid simultaneous transmissions of the newly generated data packets, each source node waits for an additional random time. The size of each data packet is 128 bytes, the maximum

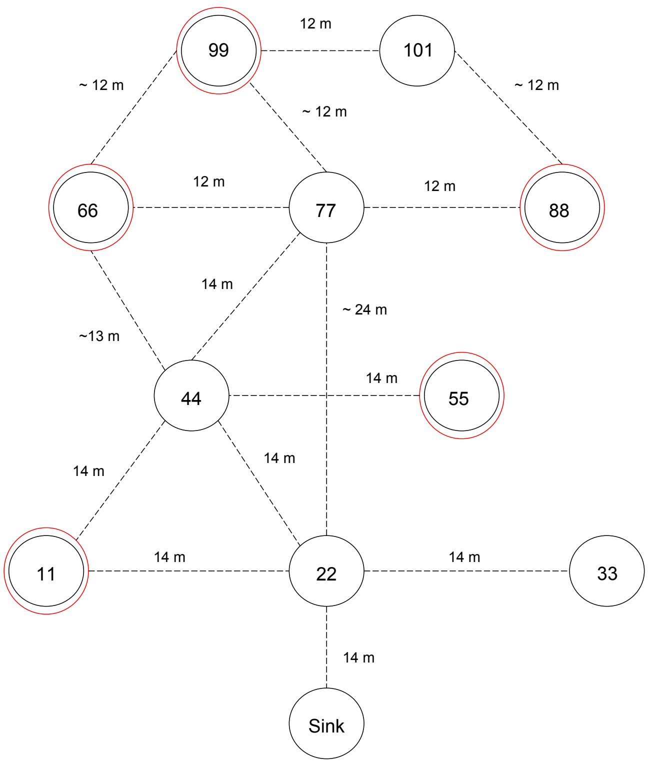


Figure 4.4: Graphical representation of the testbed.

allowed by radios compliant with the IEEE 802.15.4 standard. The data rate of the main radio is 250 kbps.

Table 4.1: Experiment parameters

Description	Value
Experiment length	150 min
Number of nodes (total)	11
Source nodes	5
Inter-packet interval	60 s
Main radio TX power	−2 dBm
Main radio bitrate	250 kbps
Wake-up TX power	10 dBm
Wake-up bitrate	1 kbps
Data packet size	128 bytes
Wake-up sequence size	32 bits
Transmission attempts (total)	10
Grandparent node	5
Parent node	5
Trickle timer	
Minimum interval τ_{min}	10 s
Maximum interval τ_{max}	500 s
LPL sleep interval	500 ms
Variant parameters	
ACKs sampling interval	5 packets
Grandparent ACKs threshold	70%

In order to have a more reliable communication over the wake-up channel, we encode the wake-up sequences with a Hamming code H(8,4). Before the encoded content of the wake-up request, 8 additional *sync bits* are sent. In total, each wake-up message is 32 bits long, and it takes 32 ms to be transmitted at 1 kbps.

Each experiment lasts 150 minutes. It can be seen as made of three different phases: an initial *setup phase* of 20 minutes during which only CTP beacons are exchanged, in order to build the routing tree. During the second phase, which lasts 120 minutes, data packets are generated and injected into the network. In the last 10 minutes of the experiment the source nodes do not generate any new data packet. This last phase allows the nodes of the network to deliver to the sink all the data packets that may have gathered in their buffers.

When using LPL each node periodically checks for channel activity. In our experiments, every node, included the sink, checks if there is an ongoing transmission every 500 ms. We decided to have the sink follow the same duty-cycle as the rest of the nodes in CTP-LPL in order to have longer routing paths than in CTP-WUR where the sink had its main radio always active. In this way we obtained an average number of hops crossed by every data packet of 1.54, for CTP-WUR and 1.78 for CTP-LPL.

We changed how CTP sends beacons. A beacon is always preceded by a wake-up sequence, which takes 32 ms to be transmitted. When the network is started, all the nodes send beacons at high frequency, in an attempt to build the routing topology as soon as possible. In this phase, the long transmission time of the wake-up sequences increases the probability of collisions. Collisions on wake-up sequences prevent the nodes to turn on their radios and receive the beacons. Some nodes may end up disconnected from the rest of the network. In order to decrease the possibility that more nodes decide to send a beacon almost at the same time, causing collisions over the wake-up sequences, we raised the minimum value τ_{min} of the Trickle timer. This allows to grow the size of the initial interval in which the nodes randomly pick their wait time before the beacon transmission. We tried several values, starting from the TinyOS default of 128 ms up to 10 seconds. Even with such a big interval, some nodes, especially those in the last rows of the testbed, still may end up disconnected. To further reduce beacon traffic during the setup phase, we forbid the nodes that still have not found the parent to solicit their neighbors to send them their routing information. When the network starts, only the sink transmits beacons. Its neighbors are the first nodes to find the parent and can subsequently start sending beacons to advertise their route to the sink. In this way, as the routing tree is gradually built, also the nodes gradually start sending beacons, from the closest to the sink to the farthest. Moreover, since every node follows its own Trickle timer (i.e., after every beacon it is doubled) the beacon transmission rate is different between the first nodes that have found the parent and the last nodes. This also decreases the chances of collisions.

The number of forwarding attempts to the grandparent, T, is set to 5. If, after these 5 transmissions attempts, the packet has still not been acknowledged the sender tries to forward it to its parent for maximum 5 times, before discarding it. Therefore, the maximum number of transmission attempts per packet is set to 10.

All the other parameters used by CTP have been left to their default values.

4.4 Performance metrics

We compare CTP-LPL against CTP-WUR and our proposed variant using the following metrics:

1. End-to-end packet latency, i.e., the time from packet generation to its correct delivery to the sink.
2. Energy consumption, i.e., the total amount of energy consumed by all the nodes of the network.
3. Main radio active time, i.e., the percentage of time spent by the main radio in the active state, relative to the experiment time.
4. Packet Delivery Ratio (PDR), i.e., the percentage of generated packets delivered to the sink.

The results are averaged over 10 runs of the experiment.

4.5 Performance results

In addition to the results of our proposed optimization of CTP-WUR we also show the results of the "standard" CTP-WUR with a higher number of retransmissions (10 to the grandparent plus 10 to the parent).

4.5.1 End-to-end latency

Figure 4.5 shows the average end-to-end packet latency of each protocol during our experiments. The optimized CTP-WUR (129.6 ms) obtains latencies that are around 8 times less than those of CTP-LPL (1025.7 ms). The basic CTP-WUR (391.6 ms) performs worse than the optimization, but still better than CTP-LPL.

The lower number of retransmissions is one of the reasons that explain why the variant of CTP-WUR has better latencies, as nodes try to forward

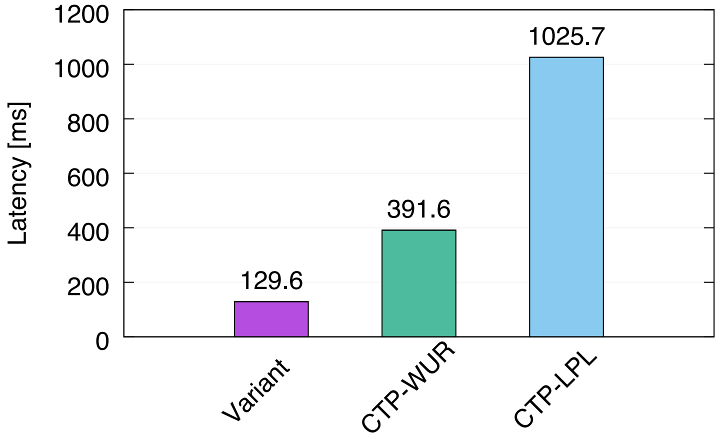


Figure 4.5: Latency

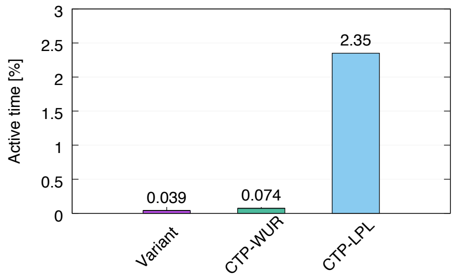
packets for shorter periods. This also results in a lower probability of collisions on the wake-up sequences sent before the data packets. The second reason is the variant itself: when a node notices that the packets it has forwarded are mostly acknowledged by its parent and not by the grandparent, it decides to stop wasting time and energy trying to send packets to a node that, at least temporarily, it cannot reach.

All versions of CTP-WUR obtain better latencies than CTP-LPL because of their use of the wake-up radio. When a node has a packet to transmit it does not have to wait for the next-hop to turn on the radio according to its duty-cycle: it just have to request its activation through the wake-up sequences. In CTP-WUR, the 1-hop latency, i.e., the time needed to send a packet to the next-hop (different from the sink), including the wake-up sequence relaying phase, is around 80 ms.

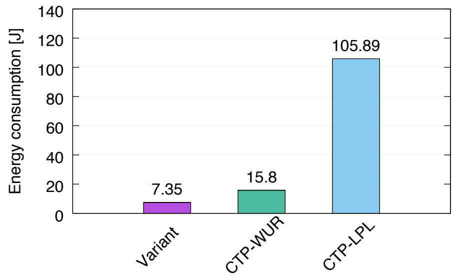
We recall that in CTP-WUR the sink always had its main radio active and listening to the channel. Therefore, when the sink is the next-hop (being it parent or grandparent) the wake-up phase is avoided. Instead during the CTP-LPL experiments, in order to have longer routing paths, the sink follows the same duty-cycle of the rest of the network.

4.5.2 Energy consumption

Figure 4.6a and Figure 4.6b show, respectively, the average percentage of active time of the main radios and the total amount of energy consumed by the network when running the evaluated protocols.



(a)



(b)

Figure 4.6: (a) Main radio's percentage of active time. (b) Energy consumption.

The energy consumption of each node takes into account the energy spent for communication (both main radio and wake-up radio) and assumes that the node is put into a low-power sleep state when the main radio is off. To measure the energy spent for communication we measured the time spent by the main radio, wake-up transmitter and wake-up receiver in active and idle states.

When the nodes run either of the CTP-WUR protocols their main radios are active for less than 0.1% of the experiment time. In particular, during the experiments with our proposed version of CTP-WUR the main radio is active for less than 0.04% of the time. Without the variant the main radio usage is about twice as much. The reason for this result is twofold: the first is that the variant is designed to avoid transmission attempts to the grandparent node when the channel conditions do not allow a good enough communication. The second one is the lower number of retransmission attempts allowed during the experiments of the variant.

These results match with those of the energy consumption of the network, because it is directly proportional to the main radio active time. In detail, our version of CTP-WUR averaged an energy consumption of 7.35 J, while CTP-WUR consumes a little bit more than twice as much, 15.8 J. With CTP-LPL the main radio usage soars at 2.35% of the experiment time, 60 times more than our variant of CTP-WUR. Accordingly the network energy consumption is much higher than that of CTP-WUR (105.89 J): 14 times more than the variant and 7 times more than the regular version of CTP-WUR. These results clearly show the effect of idle listening in CTP-

LPL and that CTP-WUR is able to remove it completely, since it turns on the main radio of the nodes only when it is necessary.

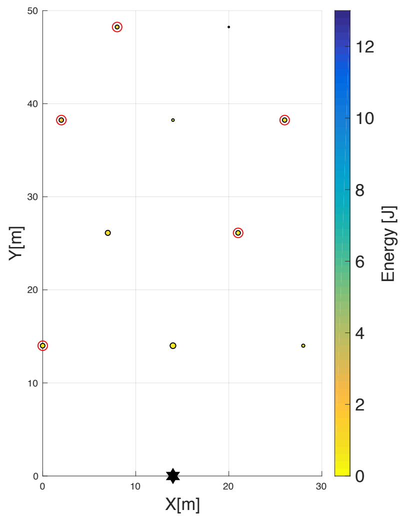
Figures 4.7a, 4.7b, 4.7c show the percentage of active time of the main radio and the energy consumption of each node of our testbed during the experiments with the optimized CTP-WUR (Figure 4.7a), the standard CTP-WUR (Figure 4.7b), and CTP-LPL (Figure 4.7c). Source nodes are highlighted by a red circle, while the black star at the bottom of the deployment area denotes the sink. The size of each node is proportional to the amount of time they used the main radio. Nodes that have their main radio active for a longer period are bigger in size. The color of the nodes indicates their energy consumption. The darker the color, the higher the energy consumed.

These figures can give an idea of which nodes the data traffic went through. Every source node generated the same amount of data packets, therefore they used the main radio for similar amounts of time. In the experiments of the non-optimized CTP-WUR source nodes 11, 55 and 66 used the main radio for about twice the time of the other two source nodes. Therefore they are bigger in the Figure. The reason is that they relayed most of the data traffic. Due to its position, right in front of the sink (see Figure 4.4), node 22 relayed almost all the data packets generated. Instead, node 101 was almost never involved in packet forwarding: its radio was active for 0.01% of the experiments time with the regular CTP-WUR and for 0.005% of the time with our optimization. Even though node 33 was the least used node in the CTP-LPL experiments, because of idle listening, its main radio was active for 1.59% of the experiment time, 159 times the value of the least used node in CTP-WUR without the variant, and 318 times the value of the least used node in CTP-WUR with the variant.

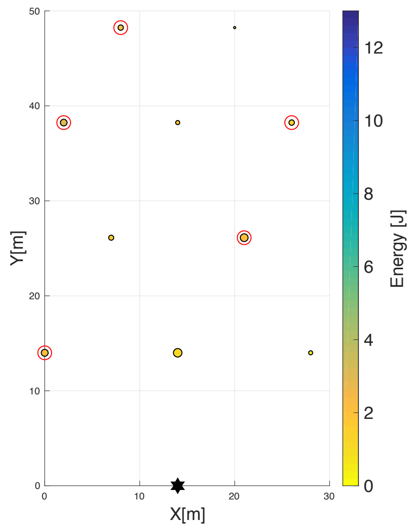
4.5.3 Packet Delivery Ratio

Figure 4.8 shows the average PDR obtained by each protocol. CTP-LPL with 99.5% of packets delivered performs slightly better than CTP-WUR with the variant (98.3%).

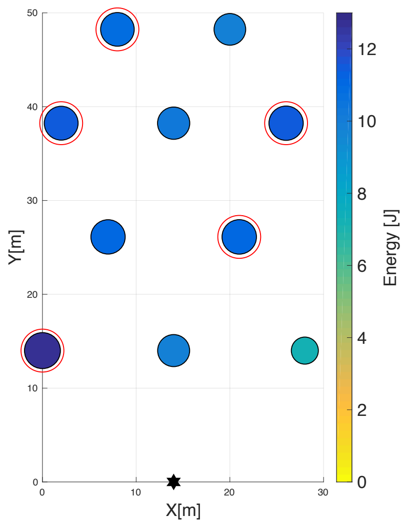
The main reason for packets being dropped by CTP-WUR was that the next-hop (being the parent or the grandparent) did not wake-up either because it received no wake-up message at all or because it received a wake-up message with too many wrong bits for the hamming encoding to correct



(a) CTP-WUR with variant



(b) CTP-WUR



(c) CTP-LPL

Figure 4.7: Main radio active time and energy consumption per node.

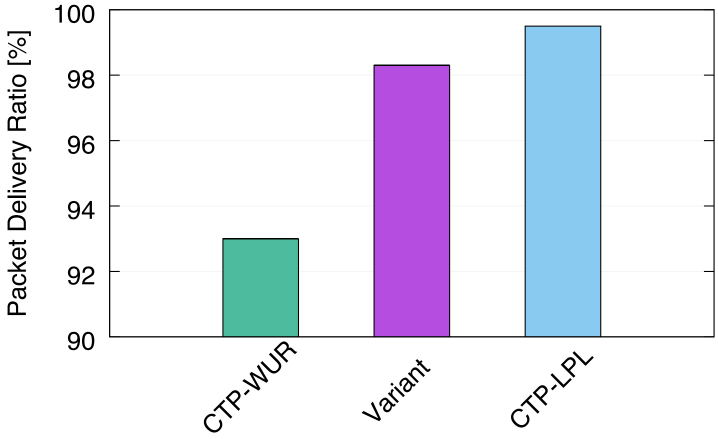


Figure 4.8: Packet Delivery Ratio

it. CTP-WUR without the optimization and with twice the number of retransmissions attempts performs worse, delivering only 93% of the packets. Instead of increasing the resiliency to errors, a higher number of retransmission attempts damages the capability of the network to deliver packets to the sink. Since every retransmission is preceded by a wake-up request, increasing the maximum number of attempts increases the probability of collisions over wake-up packets. Therefore the nodes would be less likely to wake-up and receive the data packets.

4.6 Conclusions

We proved the benefits of wake-up radio technology testing a wake-up enabled routing protocol on a testbed of MagoNode++ motes. We implemented the CTP-WUR protocol and we compared it against CTP with duty-cycle. CTP-WUR outperforms CTP-LPL both in terms of latencies (2.6 times less) and energy consumption (7 times less). We also presented an optimization of CTP-WUR that dynamically chooses whether trying to wake-up its parent node or its grandparent node in the routing tree, based on the previous packet transmissions. The goal of our proposal is to further reduce latencies and energy consumption by avoiding to transmit packets to the grandparent in case it did not receive a sufficient amount of the latest packets. The experiments show that the variant actually provides better latencies (around 10 times less than CTP-LPL) and energy consumptions

(14 times less than CTP-WUR). The only metrics in which CTP-LPL performed better is Packet Delivery Ratio: 99.5% against 98.3% of the variant of CTP-WUR and 93% of the basic CTP-WUR. The reason for this difference is that the wake-up radios have not always received a valid wake-up packet, most likely because of collisions over wake-up packets. In fact, since their transmissions take 32 ms this increases the chances that more than one node tries to wake-up its next-hop at the same time.

Chapter 5

Concluding remarks

The widespread adoption of Wireless Sensor Networks is hindered by the scarce amount of energy held by the batteries that power the network nodes. This inspired many researchers to propose protocols in which the radio is activated following a duty-cycle. The performance of these protocols, however, is not enough for a family of applications that require the networks to be operational for several years with end-to-end latencies in the order of few tens of milliseconds.

Wake-up radio technology allows to reach both these goals at the same time. Research has focused mainly on the design of new ultra-low power wake-up receiver, while protocols tailored on the features of wake-up radios have not seen the same level of productivity. Additionally, only a handful of these protocols have been implemented and evaluated on real devices.

This thesis aims at contributing to the research on energy-efficient wireless networking with the performance characterization of an ultra-low power wake-up receiver and with the implementation and evaluation on a real-world testbed of a routing protocol for wake-up radio-enabled Wireless Sensor Networks.

In Chapter 3 we investigated the range performance of an ultra-low power wake-up radio receiver prototype. Through a set of both outdoors and indoors experiments we showed that the prototype is able to receive more than 96% of the transmitted wake-up sequences when deployed indoors at a distance of 24 meters and more than 99% when used outdoors at a distance of 21 meters. Further experiments performed months later showed that the prototype operating at 868 MHz can be severely affected by interference, causing a reduction on the percentage of received sequences and percentage of True Positives.

In Chapter 4 we showed the results of the experimental evaluation on a testbed of our implementation of CTP-WUR. We introduced and evaluated also a variant of the protocol, designed to further improve packet latencies, that choose the next-hop relay based on the success of the previous packet forwardings. The comparison with the duty-cycled version of CTP showed a reduction of up to 14 times of the energy consumption of the network and a reduction of the latencies of up to 8 times.

Bibliography

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122 – 140, 2017.

[2] A. Cammarano, D. Spenza, and C. Petrioli. Energy-harvesting WSNs for structural health monitoring of underground train tunnels. In *Proceedings of IEEE INFOCOM Workshops 2013*, pages 75–76, Turin, Italy, April 2013.

[3] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Comput. Netw.*, 54(15):2688–2710, October 2010.

[4] S. Basagni, M. Y. Naderi, C. Petrioli, and D. Spenza. Wireless sensor networks with energy harvesting. In S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors, *Mobile Ad Hoc Networking: Cutting Edge Directions*, chapter 20, pages 703–736. John Wiley & Sons, Inc., Hoboken, NJ, March 5 2013.

[5] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM 2002*, volume 3, pages 1567–1576. IEEE, 2002.

[6] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In *Proceedings of ACM SenSys 2006*, pages 307–320, New York, NY, USA, 2006. ACM.

[7] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of ACM SenSys 2004*, pages 95–107, New York, NY, USA, 2004. ACM.

[8] David Moss and Philip Levis. BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking. Technical Report SING-08-00, Stanford University, Stanford, CA, 2008.

[9] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli. Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems. In *Proceedings of IEEE INFOCOM 2015*, pages 522–530, April 2015.

[10] Texas Instruments. CC1101 Datasheet. rev. SWRS061I, 2013.

[11] Crossbow. TelosB Datasheet. 6020-0094-01 Rev B.

[12] Atmel. ATmega256RFR2 Datasheet, 2014.

[13] S. Basagni, Federico Ceccarelli, F. Gattuso, and Chiara Petrioli. Demo Abstract: Abating LPL-induced Latency with Wake-up Radio Technology. In *Proceedings of ACM/IEEE IoTDI 2017*, pages 299–300, Pittsburgh, PA, USA, April, 18–21 2017.

[14] Ilker Demirkol, Cem Ersoy, and Ertan Onur. Wake-up receivers for wireless sensor networks: benefits and challenges. *IEEE Wireless Communications*, 16(4):88–96, 2009.

[15] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G.U. Gamm, and L. Reindl. Has Time Come to Switch From Duty-Cycled MAC Protocols to Wake-Up Radio for Wireless Sensor Networks? *IEEE/ACM Transactions on Networking*, 2015.

[16] IEEE. 802.11ba Task Group.

[17] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of ACM/IEEE IPSN 2013*, pages 153–165, April 2013.

[18] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli. Ultra low power wake-up radios: A hardware and networking survey. *IEEE Communications Surveys Tutorials*, 19(4):2117–2157, Fourthquarter 2017.

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

[20] 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.

[21] 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.

[22] C. Petrioli, D. Spenza, P. Tommasino, and A. Trifiletti. A novel wake-up receiver with addressing capability for wireless sensor nodes. In *Proceedings of DCoSS 2014*, pages 18–25, Marina Del Rey, USA, May 26–28 2014.

[23] G. U. Gamm, M. Sippel, M. Kostic, and L. M. Reindl. Low power wake-up receiver for wireless sensor nodes. In *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 121–126, Dec 2010.

[24] F. Sutton, B. Buchli, J. Beutel, and L. Thiele. Zippy: On-Demand Network Flooding. In *Proceedings of ACM SenSys 2015*, pages 45–58, New York, NY, USA, 2015. ACM.

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

[26] S. Oh, N. E. Roberts, and D. D. Wentzloff. A 116nw multi-band wake-up receiver with 31-bit correlator and interference rejection. In *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, pages 1–4, Sep. 2013.

[27] A. Elgani, M. Magno, F. Renzini, L. Perilli, E. F. Scarselli, A. Gnudi, R. Canegallo, G. Ricotti, and L. Benini*. Nanowatt wake-up radios: Discrete-components and integrated architectures. In *2018 25th*

IEEE International Conference on Electronics, Circuits and Systems (ICECS), pages 793–796, Dec 2018.

[28] 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.

[29] 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.

[30] EPCglobal, Inc. EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz–960MHz, ver.1.1.0, 2006.

[31] M. Paoli, Dora Spenza, Chiara Petrioli, M. Magno, and L. Benini. MagoNode++: A wake-up radio-enabled wireless sensor mote for energy-neutral applications. In *Proceedings of ACM/IEEE IPSN 2016*, pages 1–2, 11–14 April 2016.

[32] L. Bedogni, L. Bononi, R. Canegallo, F. Carbone, M. Di Felice, E. F. Scarselli, F. Montori, L. Perilli, T. S. Cinotti, and A. Trotta. Dual-mode wake-up nodes for iot monitoring applications: Measurements and algorithms. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.

[33] Austria Microsystems. AS3930 Datasheet. rev. [v1-62] 2014-Nov-27, 2014.

[34] W. Z. Zhu, T. Wu, G. Chen, C. Cassella, M. Assylbekova, M. Rinaldi, and N. McGruer. Design and fabrication of an electrostatic aln rf mems switch for near-zero power rf wake-up receivers. *IEEE Sensors Journal*, 18(24):9902–9909, Dec 2018.

[35] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, Jan 2002.

[36] S. Mahlknecht and M. Spinola Durante. Wur-mac: Energy efficient wakeup receiver based mac protocol. *IFAC Proceedings Volumes*, 42(3):79 – 83, 2009. 8th IFAC Conference on Fieldbuses and Networks in Industrial and Embedded Systems.

[37] T. Stathopoulos, M. Lukac, D. Mclntire, J. Heidemann, D. Estrin, and W. J. Kaiser. End-to-end routing for dual-radio sensor networks. In *Proceedings of IEEE INFOCOM 2007*, pages 2252–2260, Anchorage, AK, May 6–12 2007.

[38] S. Basagni, Chiara Petrioli, and Dora Spenza. CTP-WUR: The Collection Tree Protocol in Wake-up Radio WSNs for Critical Applications. In *Proceedings of IEEE ICNC 2016*, Kauai, Hawaii, Feb 2016.

[39] S. Basagni, Valerio Di Valerio, Georgia Koutsandria, and Chiara Petrioli. Wake-up Radio-enabled Routing for Green Wireless Sensor Networks. In *Proceedings of IEEE VTC 2017 Fall*, pages 1 –6, Toronto, Ontario, Canada, September, 24–27 2017.

[40] S. Basagni, Valerio Di Valerio, Georgia Koutsandria, Chiara Petrioli, and Dora Spenza. WHARP: A Wake-up Radio and Harvesting-based Forwarding Strategy for Green Wireless Networks. In *Proceedings of IEEE MASS 2017*, pages 1 – 9, Orlando, FL, USA, October, 22–25 2017.

[41] T. Kumberg, M. Schink, L.M. Reindl, and C. Schindelhauer. T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers. *Ad Hoc Networks*, 59:97 – 115, 2017.

[42] F. A. Aoudia, M. Gautier, and O. Berder. OPWUM: Opportunistic MAC Protocol Leveraging Wake-Up Receivers in WSNs. *Journal of Sensors*, 501, 2016.

[43] C. Petrioli, M. Nati, P. Casari, M. Zorzi, and S. Basagni. ALBA-R: Load-balancing geographic routing around connectivity holes in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(3):529–539, March 2014.

[44] T. Kumberg, L. Reindl, M. Moharrami, and C. Schindelhauer. Improving the performance of the cross-layer wake-up routing protocol t-rome.

In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 780–785, June 2017.

[45] Microchip. PIC12LF1552 Datasheet. DS40001674F, 2016.

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

[47] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, Maria Kazandjieva, David Moss, and Philip Levis. CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 10(1):16:1–16:49, December 2013.

[48] Ugo Colesanti and Silvia Santini. The Collection Tree Protocol for the Castalia Wireless Sensor Networks Simulator. Technical Report 729, Department of Computer Science, ETH Zurich, Zurich, Switzerland, June 2011.

[49] 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 NSDI 2004*, pages 2–2, Berkeley, CA, USA, 2004. USENIX Association.

List of Figures