A Framework for MultiModal Wireless Sensor Networks

Juan Manuel Aranda López King

April, 2020

Pontificia Universidad
JAVERIANA
Bogotá

Doctoral Dissertation

A Framework for MultiModal Wireless Sensor Networks

Juan Manuel Aranda López King

Reviewers

Prof. Dr.-Ing. Jörg Nolte
Head of Distributed Systems / Operating Systems Group
Brandenburg University of Technology (BTU), Germany

María Gabriela Calle Torres, Ph.D
Departamento de Ingeniería Eléctrica y Electrónica
Universidad del Norte, Colombia

Manuel Pérez, Ph.D
Departamento de Electrónica
Pontificia Universidad Javeriana, Colombia

Supervisors

Diego Méndez, Ph.D
Departamento de Electrónica
Pontificia Universidad Javeriana, Colombia

Henry Carrillo, Ph.D
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda, Colombia

April, 2020

Juan Manuel Aranda López King
A Framework for MultiModal Wireless Sensor Networks
Doctoral Dissertation, April, 2020
Reviewers: Prof. Dr.-Ing. Jörg Nolte María Gabriela Calle Torres, Ph.D and Manuel Pérez, Ph.D
Supervisors: Diego Méndez, Ph.D and Henry Carrillo, Ph.D

Pontificia Universidad Javeriana
Department of Engineering
Cra. 7 No. 40-62 - José Gabriel Maldonado Building
Bogotá, Colombia

Abstract

Wireless Sensor Networks (WSNs) are a widely used solution for monitoring oriented applications (e.g., water quality on watersheds, pollution monitoring in cities). These kinds of applications are characterized by the necessity of two data-reporting modes: time-driven and event-driven. The former is used mainly for continually supervising an area and the latter for event detection and tracking. By switching between both modes, a WSN can improve its energy-efficiency and event reporting latency, compared to single data-reporting schemes. We refer to those WSNs, where both data-reporting modes are required simultaneously, as *MultiModal Wireless Sensor Networks* (M2WSNs).

M2WSNs arise as a solution for the trade-off between energy savings and event reporting latency in those monitoring oriented applications where regular and emergency reporting are required simultaneously. The multimodality in these M2WSNs allows sensor nodes to perform data-reporting in two possible schemes, time-driven and event-driven, according to the circumstances, providing higher energy savings and better reporting results when compared to traditional schemes. Traditionally, sophisticated power-aware wake-up schemes have been employed to achieve energy efficiency in WSNs, such as low-duty cycling protocols using a single radio architecture. These protocols achieve good results regarding energy savings, but they suffer from idle-listening and overhearing issues, that make them not reliable for most ultra-low-power demanding applications, especially, those deployed in hostile and unattended environments. Currently, Wake-up Radio Receivers (WuRx) based protocols, under a dual-radio architecture and always-on operation, are emerging as a solution to overcome these issues, promising higher energy consumption reduction and reliability in terms of latency and packet-delivery-ratio compared to classic wake-up protocols. By combining different transceivers and reporting protocols regarding energy efficiency and reliability, multimodality in M2WSNs is achieved.

This dissertation proposes a conceptual framework for M2WSNs that integrates the goodness of both data-reporting schemes and the Wake-up Radio (WuR) paradigm–data periodicity, responsiveness, and energy-efficiency–, that might be suitable for monitoring oriented applications with low bandwidth requirements, that operates under normal circumstances and emergencies. The framework follows a layered approach, where each layer aims to

fulfill specific tasks based on its information, the functions provided by its adjacent layers, and the information resulted from the cross-layer interactions. The main contributions of this dissertation are:

- The concept of M2WSNs is introduced from the data-reporting perspective and a taxonomy of energy-efficient and responsive techniques for M2WSNs is proposed.
- An energy consumption estimation model for M2WSNs is proposed that considers the behavior and performance of wake-up protocols based on WuRx in multi-hop communications. The model is compared to traditional low-duty-cycling approaches. The results give relevant insights and strong motivations for considering multi-modal approaches in WSNs.
- M2-DaGNoS, an enhanced **M**ulti**M**odal Switching mechanism for **Da**ta **G**athering and **No**de **S**cheduling is proposed and compared against state-of-the-art switching mechanisms through an extensive emulation study. The study shows that M2-DaGNoS presented a significant performance regarding energy savings, latency and reliability– all nodes within a grid can successfully run the node scheduling and data gathering mechanisms during their whole operation that do not occur in other state-of-the-art multimodal switching mechanisms.
- A framework for M2WSNs based on the WuR paradigm is reported for monitoring oriented applications with low-bandwidth requirements, and that operates under normal circumstances and emergencies. The framework is evaluated using real motes in a laboratory-based environment. Compared to a single-radio architecture based on a low-duty-cycling technique, the framework shows better reliability in terms of the event reporting latency and packet-delivery ratio, significant energy savings when considering the "worst-case", i.e., a broadcast-based wake-up scheme with one-by-one hop data transmission and acknowledgment procedures.

Acknowledgement

First of all, I owe my total gratitude to God for giving me perseverance and strength during my doctoral studies. To my family, Sandra, José María, Francisco, and Agustín, and my parents to their patience and unconditional support. I want to thank Prof. Dr. Diego Méndez and Prof. Dr. Henry Carrillo for offering me exceptional guidance and support throughout my doctoral studies. Besides, I sincerely thank Prof. Dr. Mario Schözel for his insightful comments and ideas to enrich my research from various perspectives during and after my research stay. I also like to thank the committee members, Prof. Dr.-Ing. Jörg Nolte, Prof. Dr. María Gabriela Calle Torres, and Prof. Dr. Manuel Pérez for taking the time to be part of my committee and providing insightful input. I want to thank the Centro de Excelencia y Apropiación en Internet de las Cosas (CEA-IoT) for funding my doctoral studies and the Universidad Sergio Arboleda for giving me the time to develop my doctoral research. Finally, but not least, I would like to thank everyone in the CEA-IoT team and Universidad Sergio Arboleda.

Contents

Acronyms

AN	Assisting Node
CMnt	Continuous Monitoring
CLM	Cross-Layer Manager
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTP	Collection Tree Protocol
D.C.	Duty-Cycling
EDR	Event-Driven Reporting
FW	FeuerWhere
LDC	Low Duty-Cycling
MAC	Medium Access Control
M2-DaGNoS	MultiModal Switching mechanism for Data Gathering and Node Scheduling
M2SMs	MultiModal Switching Mechanisms
M2WSNs	MultiModal Wireless Sensor Networks
PED	Parameter-based Event Detection
RDC	Radio Duty-Cycling
SNs	Sensor Nodes
TDMA	Time-Division Multiple Access
WN	Working Node
WSN	Wireless Sensor Network

WuR	Wake-up Radio
WuRx	Wake-up Radio Receiver
WuS	Wake-up Signal

List of Figures

List of Tables

1

Introduction

A traditional *Wireless Sensor Network* (WSN) consists of a network of tiny distributed nodes with measuring, computing, and data transmission capabilities. These nodes communicate wirelessly with each other and work collaboratively to monitor physical variables (e.g., temperature, humidity, pressure, among others) in an area of interest, and perform some event tracking. Typically, these nodes periodically transmit data to a sink node that it is usually in charge of performing some pre-analysis and in-network data processing [1]. Then, the sink node sends those data to users, through a border router or gateway via Internet, as shown in Figure 1.1, for further analysis and visualization.

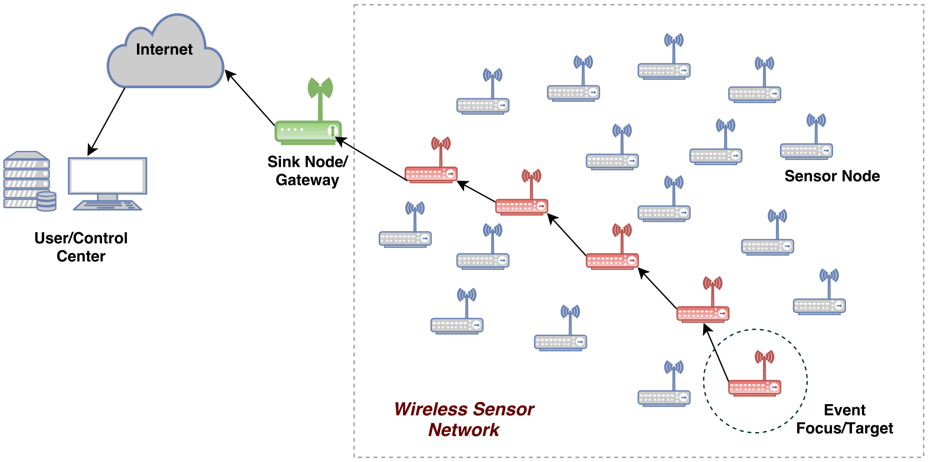


Fig. 1.1.: Typical WSN architecture for monitoring applications. The red motes represent those nodes that propagate an event and its associated data toward the sink (the green mote) via a multi-hop path. The gray motes are those redundant nodes within the network. The sink node might transmit the data packet to an observer far away from the supervising area.

Most of real monitoring and tracking applications are based on WSNs solutions. These kinds of applications are characterized by [2]:

- Wide areas of coverage (several km^2).
- Usually, low sampling rates (tenths of seconds or even minutes/hours).
- Possibility of occurrence of rapid non-periodic events.
- Battery limitations.

Besides, monitoring oriented applications can demand two type of data-reporting modes. First, a *Time-driven (proactive) reporting* or *Continuous Monitoring* (CMnt) [3, 4, 5] for supervising an area, usually under normal situations (e.g., a water quality monitoring process), where Sensor Nodes (SNs) continually monitor and report their sensed data to a sink node in a periodic fashion over time. Second, an *Event-driven (reactive) reporting* (EDR) [1] is required for event detection and tracking, usually under emergency situations (e.g., a fire event). In this mode, SNs can immediately react upon the occurrence of a predefined event, detecting abrupt changes in the value of a specific interest physical variable, and reporting to a sink node [6, 7, 8]. Those WSNs that support applications based on switching mechanisms between EDR and CMnt modes or vice versa, we named as *MultiModal Wireless Sensor Networks* (M2WSNs)[1].

1.1 Problem Statement

Nowadays, there are plenty of research around WSNs, but energy-efficiency is still a big concern for WSNs deployments [11]. In monitoring applications, especially those deployed in hostile and unattended environments [4, 5], energy saving is considered a key objective for the design of energy-efficient WSNs. Another key objective is related to the provision of delay guarantee, especially for those critical applications, in which a reliable event reporting to the sink is required (i.e., as soon as the event is detected, it should be reported to the sink for event-tracking purposes, depending on the application-context dependent) [4]. Therefore, we considered these two critical objectives to describe qualitative the trade-off that exists in M2WSNs.

Trade-off in M2WSNs

Consider an example an oil pollution monitoring application in rivers using M2WSNs (Figure 1.2). Pouring a chemical contaminant into a river–due to illegal actions of a certain factory–, might be a sporadic event with rapid dynamics, i.e., fast propagation and quick absorption by the river ecosystem [2]. A single data-reporting scheme is usually not enough for this kind of application [3]. Take, for example, a CMnt scheme in which the quality of the water is monitored regularly, every certain time, e.g., 30 minutes or 1 hour. Under normal circumstances, this scheme allows having sensed data for long periods of time, e.g., months or years. However, for an emergency such as a chemical contaminant pouring situation, the CMnt scheme requires to be configured with a higher reporting frequency (e.g., seconds or

---

[1] Some authors in the literature have introduced the concept of multimodal WSNs, but from the sensing perspective, i.e., considering two types of sensors (e.g., visual and pyroelectric infrared (PIR) [9]; PIR and metal oxide semiconductor gas sensor [10]), using only an event-driven reporting mode for event tracking or alarm propagation in the neighborhood.

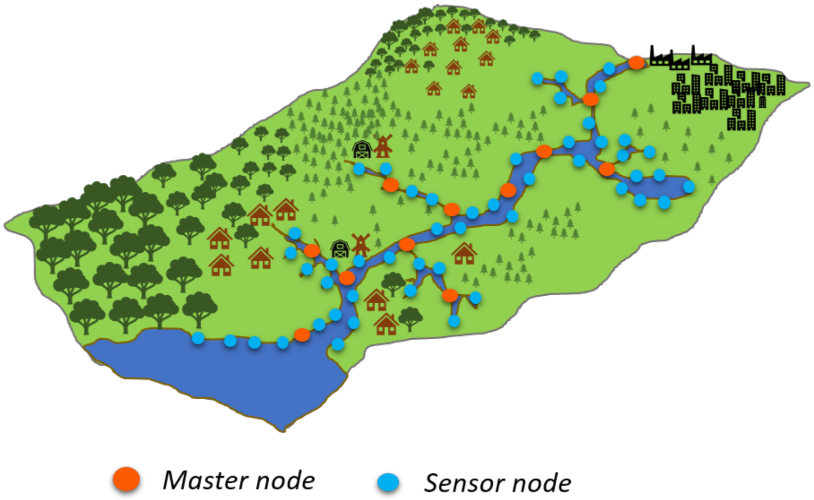


Fig. 1.2.: A development example of M2WSNs for water quality monitoring over a river (Based on [12]).

minutes) that implies a high reporting accuracy and much higher estimates of the physical variable over time, allowing to detect the event, but at the cost of higher energy consumption. Therefore, a reactive approach might be more appropriate (an EDR scheme) that reacts only and immediately upon the occurrence of an event, offering a higher reporting accuracy over short periods of time for detection and tracking purposes, but for regular monitoring might not be enough. Hence, there is a trade-off to be made between reporting accuracy and energy saving regarding these data reporting modes [6].

Under this context, we defined *reporting accuracy* as the end-to-end latency of reporting an event, or more specifically, the time between the exact occurrence of the event and the moment an observer (sink node) finally receives the first event packet. Some underlying assumptions were made with respect to this definition:

- Sensors handle two sensing modes in the detection of physical variables: a *high precision* mode for shorter sensing ranges, and a *low precision* mode for most extended sensing ranges. The sensors are able to set on the high precision sensing mode during the working and assisting operations for event detection and tracking. The sensing range is smaller than the communication range.

- The event is known, e.g., if the sensed data are outside a defined range or threshold, an event has occurred, (e.g., $T \geq 100°\text{C}$). Hence, the SNs know beforehand when to report or not an event packet to the sink.

- Given a target error, we chose a variable "delta", Δ, to meet the error. $\Delta = \{\delta_{min}, \delta_{max}\}$ defines the number of packets (granularity) reported over time towards the sink for event characterization purposes. δ_{min} implies a lot of event packets reported and higher energy consumption. δ_{max} implies fewer packets reported, and lower energy consumption. Δ is application-dependent.

- After reporting the first (critical) packet (i.e., an event occurred), the latency of the rest of data packets is defined by the communication technology (e.g., IEEE 802.15.4, MAC protocol) and the known path towards the sink.

Trade-off Formalization. The trade-off in M2WSNs can be formalized as an optimization problem, where the goal is to minimize both the total energy consumption and the end-to-end delay, given a desired reporting accuracy (i.e., ℓ–an upper-bound latency, application-dependent). Hence,

$$\begin{aligned} \text{minimize} \quad & f_1 = \sum_{i=1}^{N} E_i \\ \text{s. t.} \quad & f_2 = \sum_{j=1}^{k} D_j \leq \ell \end{aligned} \tag{1.1}$$

where E_i is the energy cost of the i-th node during its whole operation. The energy reduction is related to many aspects such as sensing, processing, switching, and communication. D_j is the one-hop latency, where $j = \{1, 2, ..., k\}$ is the individual link [4] or jth hop (node) [13] of the selected routing path towards the sink. The latency is usually affected by the mote limitations (e.g., energy source, communication, and computational capabilities), the unreliable wireless links and the large-scale nature of WSNs [4] that are also considered important constraints of the problem regarding the complexity of the algorithms running in the motes (i.e., SNs).

Solution Approach

Techniques that implement a switching mechanism between continuous monitoring and event-drive reporting modes to combine the advantages of both schemes are considered a solution approach for the trade-off mentioned above. These techniques are called as *MultiModal Switching Mechanisms* (M2SMs). A sensor node that supports M2SMs can react immediately upon emergencies, using the EDR capabilities, and offers continuous monitoring features during normal circumstances by the CMnt capabilities. In Chapter 5, we showed that by combining M2SMs with low-duty-cycling (LDC) and power-aware network techniques, the energy savings of the whole M2WSNs is further improved.

Moreover, the M2SMs implemented under a dual-radio architecture and networking techniques based on the wake-up radio and cross-layer design paradigm provide even better performances regarding latency and reliability, as shown in Chapter 6. Hence, M2WSNs achieve significant energy savings and important reporting accuracy for monitoring applications.

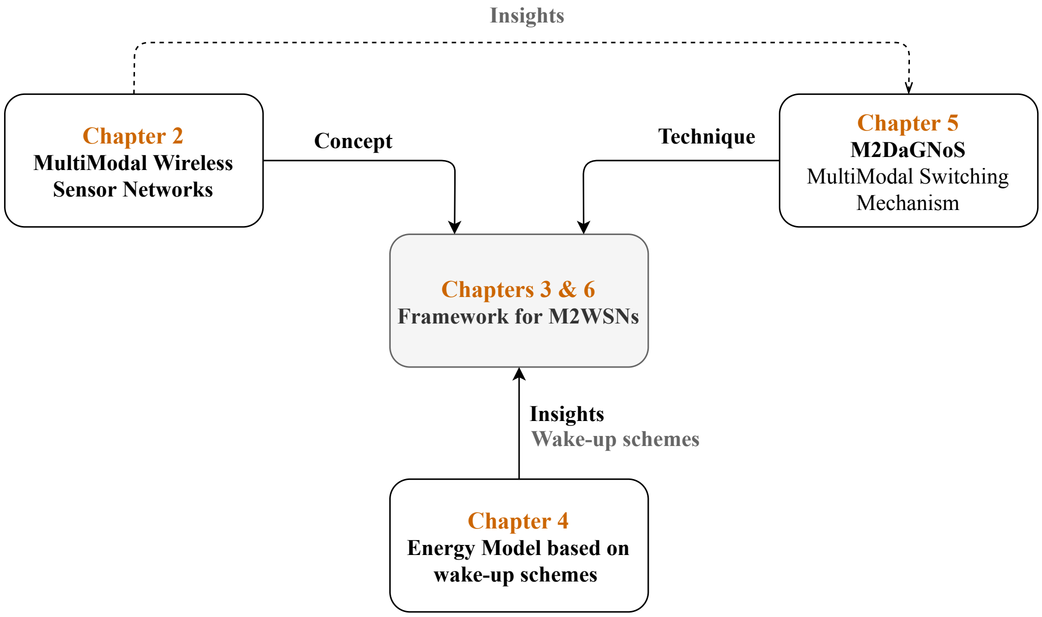


Fig. 1.3.: Overview of our contributions.

1.2 Contributions and Dissertation Structure

In this dissertation, we propose a reliable energy-efficient framework[2] for M2WSNs for monitoring oriented applications with low-bandwidth requirements, and operating under normal circumstances and emergencies. An overview of our contributions are shown in Figure 1.3 and described below, considering the dissertation structure.

Chapter 2: MultiModal Wireless Sensor Networks. We introduce the concept of M2WSNs from the perspective of data-reporting modes, and we present, from an energy-efficiency perspective, a review of switching mechanisms for M2WSNs. Besides, we explore two sophisticated techniques required in M2WSNs for further energy savings and event reporting latency reduction purposes: *Duty-cycling (D.C.)* and *Wake-up Radio (WuR)*. We highlight insights concerning switching and network management techniques for M2WSNs. It is worth to mention that this chapter is based on the following publication.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry. **MultiModal Wireless Sensor Networks for Monitoring Applications: A review**. *Journal of Circuits, Systems, and Computers*. World Scientific. Volume 29, Issue 2. 2020. pp: 1-32. ISSN:0218-1266. DOI:10.1142/S0218126620300032.

Chapter 3: A framework for M2WSNs. We give a general overview of the proposed framework for M2WSNs suitable for monitoring oriented applications with low bandwidth

---

[2] The term *Framework*, in this dissertation, refers to a layered architecture that indicates the set of techniques or protocols that can be implemented and how they would interrelate with each other [14].

requirements. The framework follows a modular or layered approach, where each layer aims to fulfill specific tasks based on its information, the functions provided by its adjacent layers, and the information resulted from the cross-layer interactions. Some techniques associated with each layer are described in this chapter, and some others will be introduced in Chapters 5 and 6. This chapter is based on the following papers.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry; SCHOELZEL, Mario. **A Framework for MultiModal Wireless Sensor Networks**. Submitted to *Ad Hoc Networks Journal*. Elsevier. 2020. ISSN: 1570-8705. DOI:-.

Chapter 4: An Energy Consumption Model based on Wake-up Schemes. We introduce an energy consumption estimation model that considers the behavior and performance of wake-up protocols based on WuRx and the traditional LDC schemes employed in M2WSNs. This model allows us to get more insights into the differences between both approaches and for considering a multimodality feature in WSNs by combining different transceivers and reporting protocols from an energy-efficiency perspective. This chapter is based on the following publications.

- ARANDA, Juan; SCHOLZEL, Mario; MENDEZ, Diego; CARRILLO, Henry. **An Energy Consumption Model for MultiModal Wireless Sensor Networks based on Wake-up Radio Receivers**. In *IEEE Colombian Conference on Communications and Computing (COLCOM2018)*. Medellin, Colombia. May 16-18, 2018. IEEE Xplore Digital Library. ISBN: 978-1-5386-6820-7. DOI: 10.1109/ColComCon.2018.8466728.
- ARANDA, Juan; SCHOELZEL, Mario; MENDEZ, Diego; CARRILLO, Henry. **MultiModal Wireless Sensor Networks based on Wake-up Radio Receivers: An analytical Model for Energy Consumption**. *redin Revista Facultad de Ingeniería – Universidad de Antioquia*. redin. No. 91. May 2019. ISSN:0120-6230. DOI:10.17533/10.17533/udea.redin.20190401.

Chapter 5: A MultiModal Switching Mechanism for Data-Gathering Schemes. We present the design, implementation and evaluation of M2-DaGNoS, an enhanced MultiModal Switching mechanism for Data Gathering and Node Scheduling, suitable for monitoring applications that simultaneously manages normal and emergency circumstances. This technique is implemented in ContikiOS, an open-source operating system for the Internet of Things, and evaluated using the Cooja emulator through an extensive simulation study. The study shows that M2-DaGNoS presents a significant performance regarding energy savings, reporting latency and reliability–all nodes within a grid can successfully run the node scheduling and data gathering mechanisms during their whole operation that does not occur in other state-of-the-art multimodal switching mechanisms. Besides, M2-DaGNoS was validated in a laboratory-based environment regarding energy consumption and event reporting latency under a single-radio architecture and a low-duty-cycling technique. This chapter is based on the following papers.

- ARANDA, Juan; CARRILLO, Henry; MENDEZ, Diego. **Enhanced Multimodal Switching Mechanisms for Node Scheduling and Data Gathering in Wireless Sensor Networks**. In *IEEE Colombian Conference on Communications and Computing (COLCOM2017)*. Cartagena, Colombia. August 16-18, 2017. IEEE Xplore Digital Library. ISBN: 978-1-5386-1060-2. DOI: 10.1109/ColComCon.2017.8088194.
- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry; SCHOELZEL, Mario. **M2-DaGNoS: a Data Gathering and Node Scheduling Mechanism for MultiModal Wireless Sensor Networks**. *Computer Communications (ComCom) Journal*. Elsevier. Vol 148. pp. 165–175. December 2019. ISSN:0140-3664. DOI:10.1016/j.comcom.2019.09.012.

Chapter 6: Framework Performance Evaluation. We present the performance evaluation results of the proposed framework for M2WSNs based on the Wake-up Radio paradigm introduced in Chapter 3. To that end, we conducted several experiments using real motes in a laboratory-based environment and compare the proposed framework against the single-radio architecture based on a low-duty-cycling technique introduced in Chapter 5. The performance evaluation results indicate that framework has better reliability in terms of the event reporting latency and packet-delivery ratio and significant energy savings when considering a broadcast-based wake-up scheme with one-by-one hop data transmission and acknowledgment procedures. This chapter is based on the following paper.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry; SCHOELZEL, Mario. **A Framework for MultiModal Wireless Sensor Networks**. Accepted for publication in *Ad Hoc Networks Journal*. Elsevier. 2020. ISSN: 1570-8705. DOI:-.

Chapter 7: Conclusions. We conclude this dissertation by providing a summary of our contributions and discuss some prospects and opportunities for the future regarding the construction of reliable and energy-efficient M2WSNs.

MultiModal WSNs: Literature Review

2

Wireless Sensor Networks (WSNs) are a widely used solution for monitoring oriented applications (e.g., water quality on watersheds, pollution monitoring in cities). These kinds of applications are characterized by the necessity of two data-reporting modes: time-driven and event-driven. The former is used mainly for continually supervising an area and the latter for event detection and tracking. By switching between both modes, WSNs can improve their energy-efficiency and event reporting latency, compared to single data-reporting schemes. We refer to those WSNs, where both data-reporting modes are required simultaneously, as M2WSNs. In this chapter, we introduce the concept of M2WSNs, and we present, from an energy-efficiency perspective, a review of switching mechanisms for M2WSNs. Besides, we explore two sophisticated techniques required in M2WSNs for further energy savings and event reporting latency reduction purposes: duty-cycling and wake-up radio. We highlight insights concerning switching and network management techniques for M2WSNs.

2.1 Overview

As mentioned in Chapter 1, most of real monitoring and tracking applications employ traditional WSNs that consist of small SNs that work collaboratively to supervise specific areas and track critical events in an energy-efficient and confident way. However, there are special WSNs in which SNs perform two data-reporting modes. First, a time-driven data-reporting mode for periodic monitoring situations in a target environment. Second, an event-driven reporting mode for keeping track of critical events upon their occurrence, with the aim of energy consumption reduction in SNs [6, 3]. We named these WSNs as *MultiModal Wireless Sensor Networks (M2WSNs)* from the data-reporting perspective.

From an application point of view, both data-reporting modes present particular characteristics:

1. *Time-driven Reporting or Continuous Monitoring mode*:
    - SNs periodically transmit their sensed data to the sink node under a normal situation.
    - It permits higher event detection accuracy, but it usually has higher energy consumption, especially when a small data-reporting period is selected (i.e., more sensed data are reported over time).

    - SNs are usually woken-up through a sleep-wakeup scheduling mechanism.

2. *Event-driven Reporting mode*:
    - SNs transmit their sensed data in a sporadic fashion upon the occurrence/detection of an event of interest (only under an emergency situation).
    - It reduces the energy consumption of a WSN, resulting in a better network lifetime, but it usually produces low precision estimates of monitored physical variables, given its aperiodic reporting strategy.
    - SNs are usually woken-up through a threshold criterion (e.g., $T > 100°C$) or a MAC protocol based on Wake-up Radio.

The M2WSNs arise as a solution for the trade-off between energy savings and event reporting latency in those monitoring oriented applications where regular and emergency reporting are required simultaneously, such as forest fires and pollution monitoring. Like traditional WSNs, M2WSNs include several small and battery powered SNs, randomly deployed in an area of interest to monitor physical variables (e.g., temperature) in a collaborative manner. Furthermore, the *multimodality* characterizes the M2WSNs, which allows SNs to perform data reporting in two schemes, switching between CMnt and EDR modes according to the circumstances, i.e., normal (regular) or emergency. Those mechanisms that allows SNs to switch between EDR and CMnt modes or vice versa, combining the advantages of both modes, we called as *MultiModal Switching* mechanisms.

The switching mechanisms grant SNs of M2WSNs the ability to react immediately upon triggering an event, e.g., under an emergency, using EDR capabilities. During the development of an event and its period of calm (i.e., under a normal situation), this mechanism allows SNs to periodically transmit up-to-date data of the event status using CMnt capabilities. In this way, M2WSNs help to achieve promising results regarding event reporting accuracy (latency) and energy-savings. The lifetime of M2WSNs can be further extended by integrating this mechanism with other energy management techniques, such as those based on D.C. and WuR approaches.

In the next sections, we provide a taxonomy of M2WSNs techniques. Then, we present a literature review of switching mechanisms proposed in the last 10 years for M2WSNs. Next, we give an overview of D.C. and WuR approaches and a comparison between recent protocols suggested in the literature and those implemented in real motes. Therefore, the main contributions of this chapter are summarized as follows: (i) A literature review of the state-of-the-art of switching mechanisms for M2WSNs is performed. (ii) A taxonomy proposal is presented based on the energy-efficient techniques used in M2WSNs designs

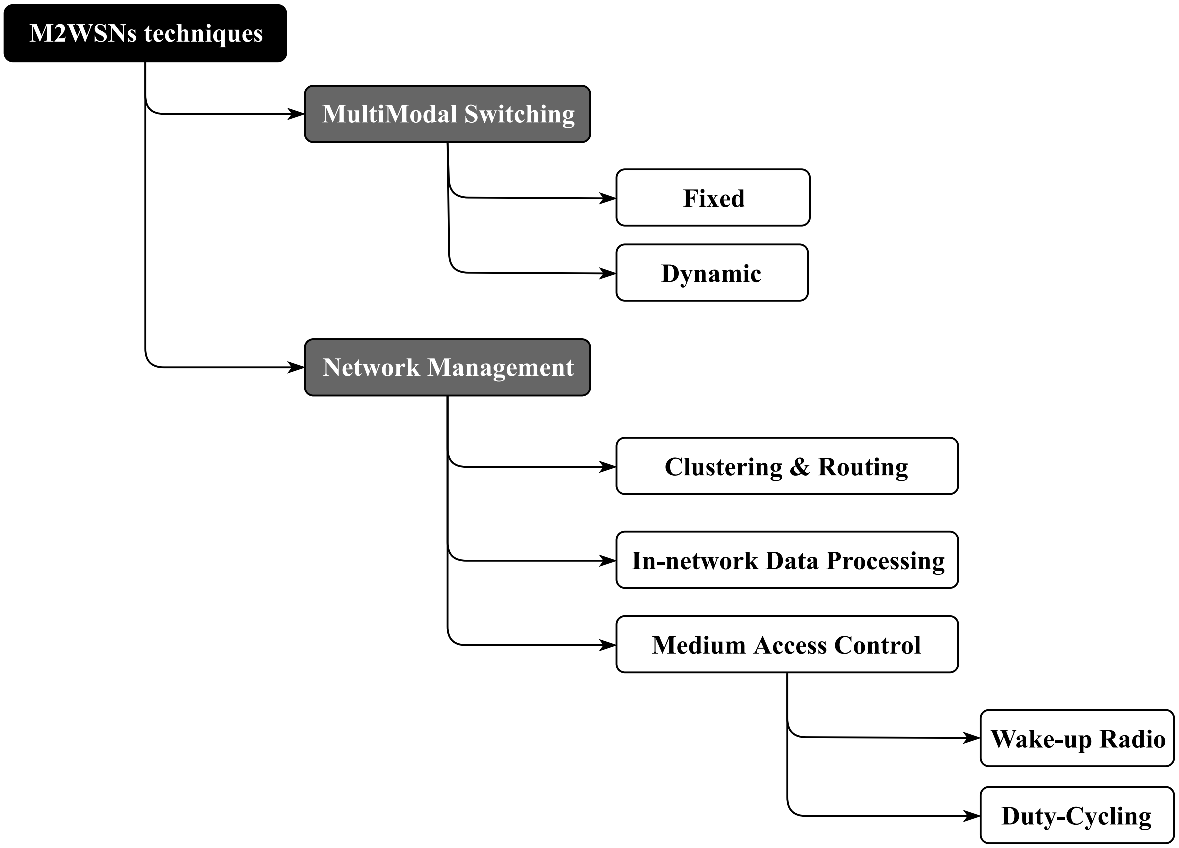


Fig. 2.1.: A taxonomy of M2WSNs techniques.

already known. (iii) A comparison is made of the D.C. and WuR mechanisms recently suggested in the literature and those implemented in real motes.

2.2 Taxonomy of M2WSNs Techniques

In this section, we propose a taxonomy of M2WSNs techniques based on the features and trade-off in M2WSNs for monitoring-oriented applications. This taxonomy has two dimensions, as shown in Figure 2.1. The *multimodality* on M2WSNs is giving basically by the first dimension–MultiModal Switching mechanisms. The M2WSNs get more energy-efficiency and reliability on delivery event-reporting packets towards the sink, through Network Management techniques–second dimension.

We focus mainly on the first dimension, and from the second dimension, those techniques based on Medium Access Control (MAC) wake-up protocols, specifically, those implemented in real hardware. We do not go in-depth into Clustering, Routing, and In-network Data Processing techniques, because they have been highly studied in the literature. Some relevant

Tab. 2.1.: Surveys and Review of Network Management techniques.

Topic	References	Description
Clustering and Routing	[15], [16]	Propose relevant techniques for grouping SNs into clusters, for network management duties, and power-aware routing protocols, for data transmission or forwarding towards the sink. These techniques are essential for reducing energy consumption during M2WSNs operations.
MAC protocols with WuR	[5], [17]	Propose a taxonomy and an in-depth analysis of relevant MAC protocols based on ultra-low WuR schemes. Besides, some routing protocols combining with MAC protocols based on this paradigm are studied. These techniques are essential for energy savings and latency reduction during a monitoring process within an M2WSN. Most of the described techniques remained in simulation studies.
MAC protocols with D.C.	[18]	Describe a taxonomy of traditional low-duty-cycling protocols proposed for energy savings in WSNs. Most of the described techniques remained in analytical and simulation studies.
In-network data processing	[19]	Describe the data aggregation and compression techniques that play an essential role in energy consumption and latency reduction in wake-up and duty-cycling WSNs. These techniques are relevant for high accuracy in monitoring processes in M2WSNs.
MultiModal switching mechanisms and Wake-up protocols	**This work**	Propose a taxonomy of M2WSNs techniques with the most relevant MultiModal Switching and Network Management techniques, focused on D.C. and WuR approaches that were implemented in real motes.

references are described in Table 2.1. Below, we provide the main features of each block presented in Figure 2.1.

MultiModal Switching Mechanisms. This dimension encompasses those mechanisms that mainly consist of switching between EDR and CMnt modes or vice versa, exploiting the capacities of both modes: energy-efficiency and high accuracy of monitoring. A comparison between EDR, CMnt and M2SMs is presented in Table 2.2. This dimension, in turn, can be classified into two groups.

Fixed: SNs switch between both modes by following a sequence of tasks during a specific interval of time (e.g., start with an EDR task, then continue with an CMnt process during 20 seconds [3]).

Tab. 2.2.: Comparison between data-reporting modes.

	EDR	CMnt	M2SMs
Data-gathering frequency	Sporadic	Periodic	Hybrid
Transmission frequency	Upon occurrence of an event	Scheduled	Dynamically adjusted
Operation schemes	Emergency	Normal	Normal & Emergency
Energy consumption[a]	Ultra-low	Higher	Low
SN's sleep-wake-up operation	WuR	D.C.	WuR & D.C.

[a] Ultra-Low (<1%), Low (1–5%), Higher (5–25%)

Dynamic: SNs switch from EDR to CMnt modes or vice versa in a sporadic fashion, according to an event behavior or an on-demand event-tracking (i.e., upon an event occurrence [6]).

Network Management Techniques. This second dimension of M2WSN techniques includes those methods widely used in WSNs for network energy management: Clustering, Routing, In-network Data Processing and MAC protocols based on D.C., and recent proposals, such as WuR [5].

Clustering: In the clustering process, SNs are grouped into clusters based on similar attributes such as geographical location and remaining residual energy value. Clustering SNs is a two-layered hierarchical architecture, where those regular SNs that sense the field form the first layer, usually known as Cluster Members (CMs). The second layer is constituted by SNs, known as Cluster Heads (CHs) that are elected among the SNs to perform duties such as gathering the data from CMs, performing data aggregation, and transmitting data to the sink nodes, usually through a multi-hop routing path among the CHs [20].

A clustering protocol approach usually passes through three phases: CH selection, cluster formation, and data transmission, where the CH selection is the main part of the protocol, that usually defines the energy-efficiency of the network. Clustering the network and selecting the CHs are typically done by either distributed or centralized mechanisms, in a simple way or in a sophisticated manner by using robust methods such as genetic fuzzy systems and fuzzy c-means algorithms [21]. In the centralized techniques, the regular SNs broadcast their location and residual energy value to the sink, then, it forms the cluster with its corresponding CH, and broadcast again to the SNs [20, 15].

A hierarchical clustering brings several benefits to the WSN management such as energy efficiency, scalability, fault-tolerance, in-network data aggregation, load balancing, high connectivity, among others [20]. The work in [15] presents a comprehensive survey of the state-of-the-art in clustering protocols for WSNs.

Routing: In recent years, several energy-efficient and energy-balanced routing protocols have been proposed in the literature. The former to increase the effective lifetime of the network by minimizing the energy consumption in each sensor node. The latter to prolong the lifetime of the network by uniformly balancing the energy consumption among the SNs in the network [16]. An extensive and comprehensive survey of energy-efficient and energy-balanced routing protocols was proposed in [16].

In-network Data Processing: This technique usually performs data aggregation, coupled with data compression techniques [1]. This mechanism exploits the correlation existing between data, through the spatial-temporal properties of SNs and data fusion, to minimize the number and size of the data packets sent towards the sink. Consequently, the data transmission cost is reduced and the energy-efficiency of the network is improved. The authors in [19] present a comprehensive survey of different aggregation and compression algorithms used for In-network Data Processing techniques.

Medium Access Control: Traditional MAC protocols are based on D.C., where SNs switch between active and sleep states, according to a scheduling mechanism, to save energy [22]. Nowadays, there are new MAC protocol proposals based on WuR that overcome some limitations of D.C. and further enhance the energy savings on SNs [5]. Typically, a WuR is periodically listening to the channel for a pre-defined Wake-up Call signal that activates, through an interrupt signal, other electronic parts of the SNs (e.g., micro-controller, communication radio) [23]. In this chapter, we focus mainly on D.C. and WuR techniques, because these techniques have been overlooked in most M2SMs recently proposed in the literature, but have great potential (Table 2.3).

2.3 MultiModal Switching Mechanisms for M2WSNs

In this section, we present a review of M2SMs proposed in the literature. A comparison table of switching mechanisms for M2WSNs is shown in Table 2.3. This comparison is made based on some relevant design criteria such as the initial data-reporting scheme (i.e., SNs can start reporting data either in a time-driven or an event-driven fashion), the switching operation mode (i.e., fixed or dynamic), the overall energy-savings in the network (i.e., low, average, high), scalability (i.e., small, medium and large density of SNs deployed in the monitoring area) and the set of techniques employed for extending the lifetime of the network. Below, each mechanism is described by presenting its novelty and weaknesses, and summarized in Table 2.4.

Bouabdallah *et al.* in [7]. They proposed CM-EDR, a continuous monitoring scheme using an event-driven reporting approach.

Tab. 2.3.: Comparison of multimodal switching mechanisms for M2WSNs.

Authors	Initial data-reporting scheme	Switching mode	Energy-savings[a]	Scalability[b]	Network techniques
Bouabdallah *et al.* [7]	EDR	Fixed	Average	Medium	MAC (CSMA), Clustering (LEACH)
Lee and Lim [6]	EDR	Dynamic	Average	Large	None
Adulyasas *et al.* [24]	CMnt	Dynamic	Average	High	MAC (CSMA), Clustering
Kang *et al.* [25]	EDR	Dynamic	High	Large	Routing (based on tier division)
Azim *et al.* [1]	EDR	Dynamic	Average	Medium	In-network Data Processing (data aggregation (SAG) and compression (CC_SCR)), Clustering (CC_SCR, LEACH), MAC (CSMA, BMAC), Routing (LEPSM)
Leyva-Mayorga *et al.* [3]	CMnt	Fixed	Average	Medium	MAC (CSMA), Clustering (LEACH)
Sun *et al.* [8]	CMnt	Dynamic	Average	Small	Routing (TDRHN)
Hu *et al.* [26]	EDR	Dynamic	High	Medium	Clustering, MAC (TDMA), In-network Data Processing (data aggregation)
Bhuiyan *et al.* [13]	EDR	Dynamic	High	Medium	MAC (TDMA, FTSP), Routing (Shortest Path), WuR (Radio-triggered Wake-up hardware)
Nagarajan and Dhanasekaran [27]	EDR	Dynamic	Average	Small	D.C.

[a] Energy-savings: Average (5-10%), High (>10%)

[b] Scalability: Small (<100 nodes), Medium (100–500 nodes), Large (>500 nodes)

Novelty: CM-EDR reports only relevant data (i.e., those data that differs from the last data transmitted) that permits a higher accuracy in data-reporting processes than only EDR schemes, and a reduction in power consumption and efficient operation when compared to only CMnt schemes. Besides, CM-EDR can be implemented over existing MAC protocols and clustering techniques, without compromising their functionalities.

An optional version of CM-EDR was proposed, OCM-EDR, that allows SNs that work as cluster heads, to go into sleep mode after several time intervals of not receiving any relevant data from its cluster members. This scheme allows to further extended the energy savings in SNs, compared to CM-EDR.

Weakness: OCM-EDR reduces the idle listening of cluster heads, and it enables a better report delay performance than CM-EDR. However, when a cluster member sensed relevant data, and its cluster head is in its sleep period, it should directly transmit this data to the sink, in order not to lose the event, but at the cost of an increase in energy consumption (if and only if the sink can be reached from the location of the sensor node).

For higher energy savings, new features can be included into OCM-EDR to decide whether a sensor node should enter on sleep mode or transmit relevant data, based on its neighbor's information. Furthermore, a WuR hardware and based protocols could be considered to enhance OCM-EDR, so that, it outcomes the CM-EDR approach. Finally, both approaches were evaluated under an analytical model, but not tested on real motes.

Lee and Lim in [6]. They proposed a hybrid data-reporting protocol that allows SNs to dynamically switch between EDR and CMnt modes, based on space and time variables.

Novelty: The hybrid protocol is based on two novel algorithms: a parameter-based event detection (PED) to determine when to switch from an EDR to a CMnt mode and vice versa, by using a threshold criteria; and a parameter-based area detection (PAD) to determine which sensor node must be involved in a continuous monitoring process. This is done by a time-to-live (TTL) propagation mechanism (similar to the one used in computer networks) and during a valid time defined beforehand. The hybrid protocol permits a higher data acquisition than an event-driven protocol, and a better energy consumption when compared to a time-driven protocol.

Weakness: The main disadvantage of hybrid data-reporting protocol is that it did not consider any additional techniques to further enhance the energy savings of a WSN, such as power-aware routing and Duty-cycling techniques. Additionally, the protocol is highly sensitive to configurable parameters that in turn depends on a target environment. Consequently, PAD algorithm could be extended to support sensor node's contextual information and a dynamically configurable parameters selection mechanism based on a target event. Finally, the protocol was not tested on real motes.

Adulyasas *et al.* in [24]. They proposed an Energy-Efficient Threshold (EET) clustering-based algorithm for event-driven data reporting within continuous monitoring schemes.

Novelty: The novelty of EET is that clusters are formed only where a data variation (i.e., an event) is occurring and during a particular time that depends on the physical phenomenon. The cluster operates as long as the duration of an event, then, the cluster will be reset and every cluster member shifts to sleep mode for energy conservation. This procedure further enhance the lifetime and stability of the network when compared to LEACH and OCM-EDR mechanisms.

Weakness: The main disadvantage of EET algorithm is that it was proved under single-hop schemes –SNs communicate directly to the sink node. Consequently, higher energy consumption is presented in large-scale WSNs. By implementing a power-aware multi-hop routing, the network lifetime can be further improved. Finally, EET was implemented in MATLAB, but it was not tested on real motes.

Kang *et al.* in [25]. They proposed a hybrid node scheduling algorithm based on an Energy-Efficient Chain (EEC) data routing approach. EEC routing works under the principle of Tier Division, according to a radio energy dissipation model.

Novelty: Though the authors presented the analytical formulation of an EEC routing, they focused mainly on the design and evaluation of two scheduling mechanisms: (1) A cyclical node sleep scheduling mechanism to keep one working node (WN) inside each grid at any time (to monitor all the grid), and turn off other nodes, to reduce energy consumption on time-driven schemes. The role of WN is rotated cyclically between nodes that join a node queue. (2) An event-driven node wake-up mechanism to wake up more nodes, so, an event coverage can be maximized, and perform precisely tracking the trends of an event. This algorithm is suitable for real complex applications where both continuous monitoring and tracking of an unusual event are required.

Weakness: The hybrid algorithm did not implement any neighbor or route discovery techniques to neighbors selection inside a grid and other grids; consequently, the data packet is not routed effectively towards the sink. Additionally, to wake up redundant nodes inside a grid for event tracking duties, a working node sends assistant messages to them, but the author did not specify what kind of wake-up mechanism was used for this purpose. The algorithm was tested under an always-on simulated scenario (i.e., nodes always listen to the communication channel). This condition did not allow significant energy savings, especially for those applications with low-power consumption requirements. The algorithm was not tested on real motes. Finally, authors assumed fully synchronized node–no drift clock issues. They did not mention any synchronization protocol.

Azim *et al.* in [1]. They proposed a set of mechanisms that aims to enhance the lifetime of WSNs by reducing the packet sizes and the numbers of packets transmitted within a communication operation in a highly correlated spatiotemporal and fairly stable environment.

Novelty: Two in-network data processing and two hybrid data-reporting mechanisms: A smart aggregation (SAG) mechanism for spatiotemporal correlation of data in CMnt applications, and a compression cluster scheme in a correlated spatial region (CC_SCR) for EDR schemes. SAG is a simple data aggregation technique that aggregates data, in an intermediate node, only if sensed data are within a predefined deviation. CC_SCR is an in-network compression technique that considers the physical characteristics of the sensed data to compress it and to elect CHs within a clustered based architecture. The compression operation consists of determining a difference between sensed values and a reference data defined by a CH. This difference is transmitted to the CHs instead of the raw data.

Tab. 2.4.: Summary of state-of-art MultiModal Switching mechanisms

Reference	Novelty	Weakness	Evaluation method	Evaluation metrics
[7]	✓ CMnt scheme using an EDR approach ✓ Run over existing MAC protocols and clustering techniques ✓ Report only relevant data (those that differ from previous data)	× Significant energy consumption due to waiting time issues × Not tested on real motes	Analytical	∗ Number of SNs alive ∗ Average energy consumption (J) ∗ Reporting latency (s)
[6]	✓ Switching based on space and time variables ✓ Parameter-based event detection (PED) algorithm ✓ Parameter-based area detection (PAD) algorithm	× Highly sensitive to configurable parameters × None additional energy saving techniques × Not tested on real motes	Simulation	∗ Total number of data transmissions in the network ∗ Ratio of SNs in a neighboring engaged to performace CMnt duties
[24]	✓ EDR within CMnt scheme ✓ Clusters are formed only where a data variation has occurred	× Only tested for single-hop communications × Not tested on real motes	Simulation	∗ Average energy dissipation (J/node) ∗ Nodes alive (%) ∗ Event detection latency (s)

Continued on next page

Tab. 2.4: Summary of state-of-art MultiModal Switching mechanisms – continued from previous page

Reference	Novelty	Weakness	Evaluation method	Evaluation metrics
[25]	✓ Energy savings by scheduling only one node in CMnt mode (per grid) ✓ Event tracking duties by waking up redundant nodes within a grid	× None neighbor and route discovery procedures × Always-on communication × Not tested on real motes	Simulation	∗ Network coverage (%) ∗ Network effective lifetime (s)
[1]	✓ Data aggregation in CMnt schemes and data compression in EDR operations. ✓ Non-Preemptive and Preemptive switching between EDR and CMnt modes	× Data aggregation and compression techniques were not integrated with the proposed switching mechanisms × Not tested on real motes	Simulation	∗ Average energy consumption (J) ∗ Number of SNs alive
[3]	✓ CMnt and EDR duties are performed in clusters formed at the beginning of operations ✓ Prioritize event packets over CMnt packets ✓ Non-Preemptive switching between EDR and CMnt modes	× Only tested for single-hop and always-on communications × Not tested on real motes	Analytical	∗ Active nodes ∗ Average event detection and report delay (s) ∗ Average energy consumption per event transmission (J)

Continued on next page

Tab. 2.4: Summary of state-of-art MultiModal Switching mechanisms – continued from previous page

Reference	Novelty	Weakness	Evaluation method	Evaluation metrics
[8]	✓ Dynamic adjustment of the data transmission rate based on threshold criteria ✓ The route selection strategy changes dynamically based on the residual energy and expected transmission value during an emergency circumstance	× Only tested assuming always-on communications × Decision parameters are selected once at the beginning of operations without considering the behavior of the emergency	Testbed	∗ Average Voltage (V) ∗ Number of SNs alive
[26]	✓ 2-logical-overlapping clustering scheme for event monitoring within a CMnt operation ✓ Switching based on the quality of services requirements (a balance between energy-efficiency and detection accuracy)	× Extra energy consumption due to synchronization duties × Only tested assuming always-on communications × Not tested on real motes	Simulation	∗ Number of SNs alive ∗ Average residual energy ∗ Network lifetime

Continued on next page

Tab. 2.4: Summary of state-of-art MultiModal Switching mechanisms – continued from previous page

Reference	Novelty	Weakness	Evaluation method	Evaluation metrics
[13]	✓ Autonomous adaptive data acquisition and switching based on the analysis of frequency contents using signal processing tools ✓ Data reduction during sampling, decision-making, and data transmission periods	× Need of a robust power-aware routing mechanism × Extra energy consumption due to synchronization duties × Troubles in decision-making duties when the highest frequency contents remain constant over long periods of time	Simualtion & Testbed	* Energy cost (mAh) * System lifetime * Energy saving in the network (%)
[27]	✓ Switching based on space and time variables ✓ Data transmission based on duty-cycling periods	× Highly sensitive to configurable parameters × Only tested in single-hop communications	Simulation & prototyping	* Power loss in percentage * Transmission percentage

NPER (Non-Preemptive Event Reporting) and PER (Preemptive Event Reporting) were proposed for efficient data transmission in schemes where both CntM and EDR are implemented simultaneously. Both mechanisms differ in the event reporting operation. NPER transmits event-data at the end of a continuous monitoring period, while PER, at the beginning of each TDMA slot.

Weakness: The mechanisms aforementioned were proposed as separate techniques, and not as an integrated framework. For instance, CC_SCR, as a compression clustering technique for EDR applications, is not used by PER and NPER. Instead, the authors implemented LEACH, presenting an inferior performance.

SAG and CC_SCR perform data aggregation, but authors did not say how the aggregation was done, e.g., by taking the average, or the median of the transmitted data. Besides, the authors assumed that nodes perform the data compression in an instantaneous manner–not the case in real deployments. Finally, these mechanisms were not tested on real motes.

Leyva-Mayorga *et al.* in [3]. They proposed WALTER (WSN Alternating CntM/ED Block Protocol for Nonpreemptive Event Reporting), a priority-based nonpreemptive hybrid WSN protocol that performs both Continuous Monitoring (CntM) and Event-Detection-Driven (EDD) duties within the same continuous monitoring cluster.

Novelty: The novelty of WALTER protocol is that it performs both time-driven and event-driven data-reporting within the cluster already formed. This means no additional energy drainage compare to separate clusters (i.e., one for EDD, and another for CntM), prioritizing event data over continuous monitoring data.

Weakness: WALTER works well for single-hop transmissions, but a power-aware routing protocol could be included to support multi-hop transmissions between CHs (to reach the base station) that is required in most large applications in M2WSN. The protocol was not tested on real motes.

Sun *et al.* in [8]. They proposed TDRHN, a Threshold sensitive Dynamic Responsive Hybrid Network (i.e., that combines proactive and reactive networks) protocol based on the Collection Tree Protocol (CTP) [28], for monitoring applications that work in normal environments and under burst situations.

Novelty: TDRHN works like a time-driven based protocol under normal circumstances, i.e., periodically reports data to the sink, keeping the data transmission frequency low for energy savings. Besides, under an emergency situation, TDRHN can dynamically adjust the sensor node's transmission data rate (increase their transmission frequency), using threshold criteria. Furthermore, TDRHN dynamically changes the routing selection strategy when a sensor node is monitoring an emergency, by considering a balance between the sensor node's residual energy and the Expected Transmission Value (ETX). Therefore, a sensor node is not only able to guarantee a reliable communication, but also, an effective energy consumption. Finally, this protocol was tested using a proof-of-concept experiment based on real TelosB motes.

Weakness: TDRHN employs an absolute and relative threshold values to adapt the transmission frequency of a sensor node dynamically. These two values are set once before SNs begin to monitor. Besides, the transmission data rate changes between two pre-defined rates, i.e., N times and $2N$ times the original rate, where N is fixed and application-dependent. In this way, the frequency of transmitting data is selected without considering the real behavior of the emergency. Consequently, the energy of SNs could be dissipated more quickly as expected, due to the higher frequency employed.

Additionally, the strategies used by TDRHN could work well for specific scenarios where all SNs in the topology and the selected routing path are always-on, but, it might fail on those topologies based-on wake-up mechanisms, where not all SNs are in their active mode to receive and transmit the packet to the next hop along the routing path. Therefore, TDRHN can be modified based on CTP-WuR approach [29], to overcome this issue and to enhance the lifetime of the network further.

Hu *et al.* in [26]. They proposed EEAOC, an energy-efficient adaptive overlapping clustering mechanism for dynamic continuous monitoring applications with some event fluctuations in cluster-based WSNs.

Novelty: The EEAOC mechanism can operate through a 2-logical-overlapping clustering scheme that encompasses non-overlapping clusters formed by traditional cluster schemes, and overlapping clusters established by each node in the WSNs as a result of choosing its next-closest cluster head. Also, a collaborative CH re-adjustment and a cluster migration techniques were designed to ensure an appropriate cluster formation during an event development inside a continuous monitoring area, by switching the proper clusters.

As Kang *et al.* in [25] mechanism, EEAOC protocol allows CHs rotation without changing the 2-logical-overlapping clustering scheme–for intra-cluster energy savings purpose. Furthermore, a hybrid data reporting strategy is operated by switching between time-driven and event-driven schemes based on the quality of services requirements that allows a good balance between the energy-efficiency and detection accuracy in dynamic continuous monitoring applications.

Weakness: EEAOC requires SNs to be synchronized so that they can properly communicate during the CHs re-adjustment and cluster migration processes. This synchronization implies that SNs need to continually exchange control messages to keep each other synchronized that causes extra traffic flow within the network, which in turn leads to additional delay and packet loss, that could compromise the quality of monitoring.

EEAOC operates under the assumption that all SNs are active during the event development; therefore, EEAOC may not be energy-effective for ultra power-demand applications. By implementing a power-aware routing, a D.C. or WuR MAC based technique, the network lifetime can be further improved.

There are few details of the design and operation of the hybrid data reporting strategy within the EEAOC mechanism. Finally, EEAOC was evaluated in MATLAB, but it was not tested on real motes.

Bhuiyan *et al.* in [13]. They proposed e-sampling, an autonomous adaptive data acquisition and low-cost monitoring scheme for WSNs applications with high-frequency and dynamic physical events. e-sampling design is focused on event-sensitive data and data acquisition optimization.

Novelty: e-sampling allows SNs to adjust their sampling rates, between low and high-frequency intervals, based on their own collected data, analyzing and making decisions on the event identification and computing their rates and intervals in a decentralized manner, i.e., without relying on their neighboring and the sink mediation. SNs performance these duties by capturing the highest frequency content and analyzing it, using Wavelet Packet Decomposition and filters–signal processing tools. Afterward, SNs estimate an appropriate sampling rate and extract an event indication of the absence or presence of an event. Finally, SNs report to the sink an event identification only if it was truly detected.

Therefore, e-sampling helps to reduce the energy cost of SNs through data reduction during sampling, decision-making, and data transmission periods; consequently, the lifetime of the network is further extended. e-sampling was evaluated in a laboratory environment using Imote2 sensor platforms and radio-triggered wake-up with synchronization modules.

Weakness: For evaluation purposes, e-sampling was implemented using the Shortest Path routing model for routing data packets towards the sink. This model might not be suitable for topologies based on Wake-up Radio and power-demand applications in M2WSNs.

Additionally, e-sampling is based on time synchronization for hop-by-hop pairwise sharing between neighboring SNs that usually requires control messages to keep SNs synchronized. Thus, the extra traffic is injected into the network, increasing the probability of a congestion situation. Consequently, the quality of monitoring and the energy-efficiency of the network might be compromised by the additional delay and packet losses generated.

Finally, e-sampling was conceived for high data rate sensing applications, where the highest frequency content captured by sensors are usually very variable over short periods of time with quickly disappearing. Thus, e-sampling might not be appropriated for applications where the highest frequency content remains constant over long periods of time, such as oil spilling and gas leaking events.

Nagarajan and Dhanasekaran in [27]. They proposed two hybrid data transmission approaches: time-driven and duty-cycling, for critical and regular WSNs applications, respectively.

Novelty: The time-driven based hybrid method is similar to the one proposed in [6]. The switching modes are managed by the PED algorithm, and upon the occurrence of an event, the event coverage area is determined by the PAD algorithm. The novelty, compare to [6], is in the duty-cycling approach, where the radio module is duty cycled for further power optimization purpose, i.e., the data transmission is subject to a fixed duty-cycling period, without compromising the data accuracy.

Weakness: Both approaches are based on PED and PAD[6]. Consequently, both are highly sensitive to configurable parameters and the target event to control. The authors tested their approaches under simulation runs (using MATLAB) and utilizing a simple prototype model under a single-hop topology, without considering any power-aware routing mechanism, common in multi-hop data transmission schemes for monitoring-oriented applications.

2.4 Network Management Techniques for M2WSNs

In this section, we present an overview of MAC protocols based-on D.C. and WuR approaches that can be applied as network management techniques, to reduce the energy consumption in monitoring applications using M2WSNs.

2.4.1 Duty-cycling

D.C. is a traditional mechanism for energy savings in most WSNs applications (e.g., environmental monitoring applications), also known as sensor scheduling [30], especially on those that require several months/years of operation before replacing or recharging the nodes' battery (e.g., a time period of more than a year–application dependent)[18]. It consists of switching ON and OFF SNs (usually their main communication radio to perform either packet reception or transmission) by scheduling their active and sleep modes in a periodic fashion, normally using sleep-wakeup schedule mechanisms. State-of-the-art of sleep-wakeup scheduling mechanisms are detailed on [31].

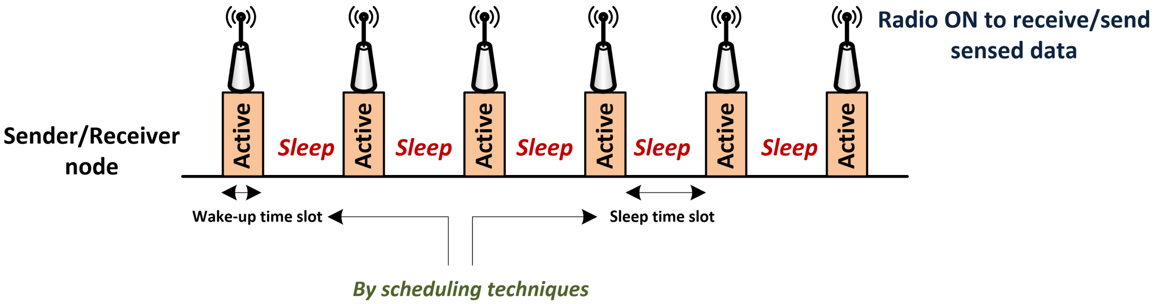


Fig. 2.2.: D.C. basic operation in WSNs. (Based on [43]). The orange block indicates the period that the radio remains active. The white spaces, the periods when the radio remains in sleeping mode. The scheduling techniques are employed to manage the radio duty-cycling over time.

D.C. Basic Operation

In Figure 2.2, the basic operation of D.C. is shown, where SNs periodically turn ON their main radio for a predefined wake-up time slot, according to a sleep-wakeup scheduling mechanism. D.C. is typically implemented in conjunction with other techniques, to prolong the lifetime of the network. For instance, wake-up scheduling [31], synchronization [32, 33], data communication [34], clustering [20] with self-healing and self-organizing features [35], and data-aggregation [36]. These techniques are also applied to solve routing problems [37, 38, 39, 40], coverage and connectivity issues [41] on WSNs.

Fixed and Dynamic D.C.

There are two types of D.C. operations: *fixed*, SNs operate with predefined duty-cycle; and *variable*, the duty-cycle is adjusted on demand, depending on a predefined trigger condition (e.g., event occurrence, traffic-load, remaining energy) [42].

Synchronous and Asynchronous D.C.

D.C. can be roughly classified into *synchronous* and *asynchronous*, based on the mechanism used to coordinate the SNs' schedules, along with some hybrid combinations [18]. In synchronous schemes, all SNs are time synchronized, having a common time base. Meanwhile, SNs on asynchronous schemes do not need to maintain a common clock.

For large-scale WSNs, asynchronous mechanisms are preferred to synchronous, due to a large amount of energy savings achieved during SNs operations [44], to be conceptually distributed and further dynamic [4]. Besides, synchronous D.C. techniques usually need a global multi-hop time synchronization, generating massive network control overheads and scalability issues [45]. Also, these techniques require fully synchronized nodes that are costly and present high implementation complexity in certain situations [46].

Asynchronous D.C. protocols present some issues that should be considered when designing a WSN. Typically, the main issue is the cumulative end-to-end waiting delay [45] that is the most critical factor for real-time event-driven applications. This issue is widely discussed in [4]. A detailed taxonomy of D.C. approaches is presented in [18].

Synchronous D.C. Techniques

Critical monitoring applications using M2WSNs may need to apply some synchronous D.C. techniques that implies synchronization mechanisms. The time synchronization problem in WSNs is not a new research topic, but it still remains as a difficult issue to be solved. Swain *et al.* in [32] presented a survey on recent clock synchronization protocols in WSNs. Most protocols have remained in simulation environments and have not been implemented in real SNs.

Djenauri *et al.* in [33] made a review on the relevant implementations of synchronization protocols, but not all protocols were thought for energy-efficient WSNs. Therefore, we extended this work by adding other protocols based on D.C., such as TAS-MAC [47] and RPL-BMARQ [48]. The most relevant synchronous D.C. techniques tested in real SNs are compared in Table 2.5.

Asynchronous D.C. Techniques

Typically, synchronous D.C. techniques are used to synchronize SNs inside clusters for continuous monitoring purposes. Large-scale environmental monitoring applications may encompass more than one cluster that, lately, interacts with other clusters asynchronously. Nevertheless, asynchronous D.C. mechanisms are preferred over synchronous for large-scale monitoring applications, because of the vast amount of energy saved during SNs operations.

Similar to synchronous D.C. mechanisms, we conducted a literature review of implemented asynchronous D.C. mechanisms, where we found X-MAC [49] and ContikiMAC [50] as one of the most used protocols. An in-depth study of ContikiMAC protocol was performed in [51]. The results showed that ContikiMAC provides better performance than X-MAC and its variants: lower latency and duty-cycle (i.e., lower energy consumption), and a higher packet delivery ratio (PDR) (i.e., fewer retransmissions). An enhanced adaptive radio D.C. version of Contiki (Contiki-AMAC) was proposed in [52].

Tab. 2.5.: Comparison of implemented D.C. mechanisms.

Mechanism	Operation/Scheme	Category-approach	Duty-cycle[a]	Data transmission	Software/Hardware
Glossy[55]	Fixed, Synchronous	Receiver-initiated	Ultra-low	Multi-hop	Contiki/T-mote sky
A+MAC[42]	Variable, Synchronous	Receiver-initiated	Ultra-low	Multi-hop	TinyOS/Mica2
GES-MAC[56]	Fixed, Synchronous	Receiver-initiated	Average	Multi-hop	Based on Mica2
RAPTS[57]	Fixed, Synchronous	Sender-initiated	Ultra-low	Multi-hop	Propietary
BailighPulse[58]	Fixed, Synchronous	Sender-initiated	Ultra-low	Multi-hop	TinyOS/TelosB
Guard beacon[59]	Fixed, Synchronous	Sender-initiated	Low	Multi-hop	Contiki/T-mote sky
L2[60]	Fixed, Synchronous	Sender-initiated	Low	Multi-hop	TinyOS/TelosB
TAS-MAC[47]	Variable, Synchronous	Receiver-initiated	Low	Multi-hop	TinyOS/TelosB
RPL-BMARQ[48]	Fixed, Synchronous	Receiver-initiated	Low	Multi-hop	Contiki/TelosB
X-MAC[49]	Fixed, Asynchronous	Sender-initiated	Low	Multi-hop	Contiki/Zolertia Z1
CONTIKI-MAC[50]	Variable, Asynchronous	Sender-initiated	Very-Low	Multi-hop	Contiki/Zolertia Z1
AS-MAC[61]	Fixed, Asynchronous	Sender-initiated	Low	Multi-hop	TinyOS/MicaZ& TelosB
MCAS-MAC[54]	Fixed, Asynchronous	Sender-initiated	Low	Multi-hop	TinyOS/Mica2
L-MAC [46]	Variable, Asynchronous	Receiver-initiated	Very-Low	Single-hop	TinyOS/TelosB
DuoMAC[53]	Variable, Asynchronous	Sender-initiated	Low	Multi-hop	TinyOS/MicaZ

[a] Duty-cycle: Average (10–25%), Low (1–10%), Very low (0.1–1%), Ultra-low (0.01—0.1%)

Recently, new variable and fixed asynchronous D.C. mechanisms for WSNs have been proposed, such as DuoMAC [53] and MCAS-MAC [54] for multi-hop transmission scenarios. The most relevant asynchronous D.C techniques implemented in real motes are compared in Table 2.5.

D.C. Challenges

Compared to an always-on approach (i.e., SNs are continually listening to the communication channel), D.C. mechanisms present a better performance in terms of energy savings, aiming at reducing *idle listening* (i.e., leaving sensor node's communication transceiver active in vain, waiting for a packet to arrive) and *overhearing* (i.e., listening to uninteresting packet wasting power in vain) issues. Nevertheless, the idle listening still constitutes a significant contributor to the overall energy consumption of SNs that also increases the data latency in the WSN [17].

A common assumption adopted in the literature is that SNs can switch from sleep to active mode at any time to transmit an available data packet, but must be awake to receive it [34]. Most of the D.C. approaches suffer from additional implicit *data latency*, due to the *waiting time*–i.e., no information is neither transmitted nor received until neighboring nodes switch to active mode [62].

Other challenges in D.C., presented in [18], are related with *collision rates* (i.e., short intervals of transmissions concentration) and *control packet overheads* (i.e., when time synchronization is needed, D.C. techniques may require to control messages). Therefore, the current energy consumption reduction achieved by D.C. techniques might be insufficient for new ultra-low-power demand designs [62].

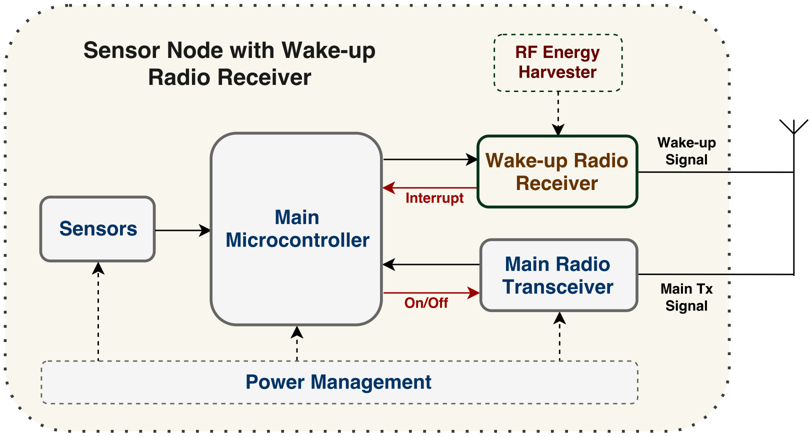


Fig. 2.3.: A generic sensor node with a WuRx hardware (Based on [17]).

2.4.2 Wake-up Radio

As a good alternative for D.C. drawbacks, the WuR approach is considered. The WuR is a hardware component, an ultra-low-power radio that is usually added to SNs as a second radio (see Figure 2.3). Some recent WuR prototypes are detailed in [63, 64, 65, 66, 67].

Typically, the energy consumption of a WuR is several orders of magnitude lower than a traditional low-power radio (for data communication), e.g., $11nA$[68], $17.5mA$[69], respectively. This condition leads to keeping the WuR always-on [17]. Consequently, WuR can be continuously sensing the communication channel, causing a reduction in data latency (waiting time), and providing energy savings and less complicated software implementation when compared to D.C. approaches [62]. Additionally, some WuR implementations have dedicated circuitry to perform an addressing mechanism by decoding a destination address contained in the packet header. Therefore, only the target sensor node is woken up rather the entire neighborhood. This feature might allow solving the overhearing issue presented in D.C. mechanisms [17].

WuR basic Operation

A WuR system typically works as follow: The main sensor node's communication transceiver and microcontroller (MCU) are in sleep mode (i.e., turned off) for energy conservation. Meanwhile, the WuR receiver (WuRx) remains active to monitor the communication channel for any eventuality continuously.

When a sensor node needs to communicate a data packet, it first sends a Wake-up Signal (WuS) through its WuR transceiver (WuTx). The remote node receives the WuS and generates an interrupt signal to wake up the sensor node's MCU, which in turn activates the main

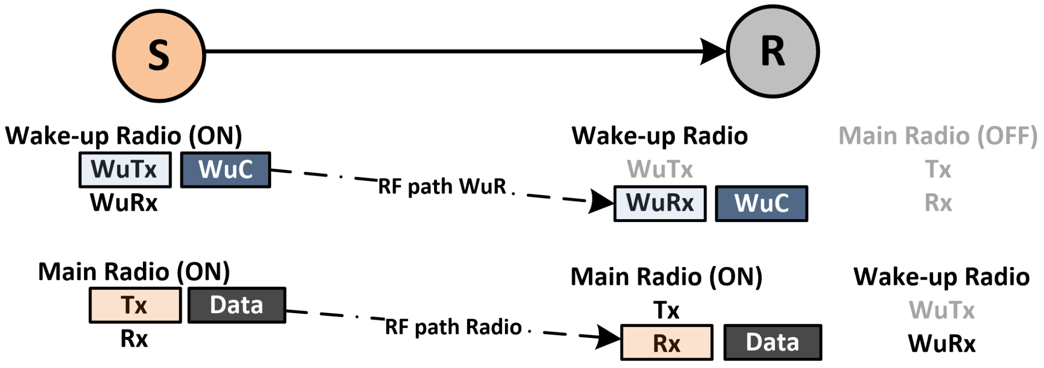


Fig. 2.4.: Wake-up Radio basic operation in WSNs under a sender-initiated approach (Based on [62]).

transceiver to receive the upcoming data. These wake-up operations help to overcome the overhearing and idle-listening issues presented in D.C. approaches [70]. After the MCU performs all the required tasks (e.g., receiving the upcoming data (using a sender-initiator WuR approach) or getting a measurement from the sensor and transmitting it back (using a receiver-initiator WuR approach)), it disables the main transceiver and goes back to an ultra-low-power mode [62]. Figure 2.4 shows the basic working principles of WuR mechanisms.

WuR based Techniques

An ultra-low-power WuR is considered a good alternative for the D.C. approach drawbacks, especially for energy conservation and latency constraints in critical event reporting applications. Recently, Djiroun and Djenouri in [5] made a complete review of MAC protocols using the WuR approach. They presented a new taxonomy and a comparative study in MAC protocols with WuR from energy and delay perspectives. They claim that MAC protocols with WuR eliminate idle listening and waiting for time problems that usually results in high energy consumption and latency. They concluded that asynchronous wake-up MAC protocols perform better than all MAC protocols that employ only one radio concerning energy-efficiency and latency requirements.

Similar to [5], Piyare *et al.* in [17] made a comprehensive overview of WuR based on MAC protocols, but they also offered an extended taxonomy of WuR based on routing protocols. They concluded that the lifetime of the SNs could be further extended by combining WuR capabilities with selective addressing and routing duties while meeting latency requirements comparable to SNs that use a single radio.

Most of the WuR based protocols remain primarily, like D.C. protocols, on simulation results without including results from real hardware implementations. During the last decade,

Tab. 2.6.: Comparison of implemented WuR protocols (Based on [17]).

Protocol	Category-approach	WuR position	Power management	Software/Hardware
WRTA[72]	Bidirectional	Receiver & Transmitter	Always-on WuR	Proprietary/Passive WuR (IEEE 802.15.4) + ST MSP430
CL-RW[73]	Bidirectional	Receiver & Transmitter	Always-on WuR	Proprietary/Active WuR (-45 dBm, 2.4Ghz) + TI LMV221, TI MB954
MH-REACH[74]	Bidirectional	Receiver & Transmitter	Energy harvesting	Proprietary/Passive WuR (915Mhz, 11.2m) + AS3992 board, TI TPS2560DRC
ZIPPY[75]	Bidirectional	Receiver & Transmitter	Always-on WuR	Proprietary/Passive WuR (-52dBm, 434Mhz, 30m) + AS3930 board, TI MSP430FR5969
WUR RPL[76]	Bidirectional	Receiver & Transmitter	Always-on WuR	Propietary/Passive WuR (21.5Khz, 3 to 5m) + Redbee Econotag mote + AS3931 board
BATS[71]	Receiver-initiator	Transmitter only	Duty-cyling WuR	Proprietary/Passive WuR (-68dBm, 868Mhz, 50m) + TI CC430/CC2590, FRDM-KL02Z
T-ROME[22]	Bidirectional	Receiver & Transmitter	Always-on WuR	Proprietary/Passive WuR (+10dBm, 868Mhz, 40m) + TI CC1101 + AS3930 board, EFM32G222F128

some WuR based protocols were evaluated on testbeds such as BATS [71] and T-ROME [22], with D.C. and always-on scenarios, respectively [17, 70]. Table 2.6 presents a qualitative comparison of relevant techniques based on WuR.

WuR vs D.C.

Figure 2.5 shows a qualitative comparison between D.C. and WuR approaches for energy management on WSNs, where WuR overpasses D.C. approach regarding advantages for ultra-low-power WSNs applications. The advantages presented for D.C. result when comparing it with an always-on mechanism. The WuRx advantages are compared to an D.C. scheme: Less complicated software [5], lower latency–WuR($<0.1s$), D.C.($>2.5s$); Higher Packet Delivery Rate–WuR($>85\%$), D.C.($<30\%$); and substantial energy-savings–WuR($<2mW$), D.C.($>9mW$)[1]. M2WSNs should consider the main advantages of both approaches to develop a robust and reliable mechanism for energy management in monitoring applications.

WuR Challenges

WuR systems need an extra hardware development, constituting a disadvantage when compared to D.C. approaches, concerning cost and energy-efficiency. Furthermore, the

---

[1] The quantitative results have been taken from the simulation study performed in [62] for a packet generation period set to $5s$ in a multi-hop static scenario (3 hops).

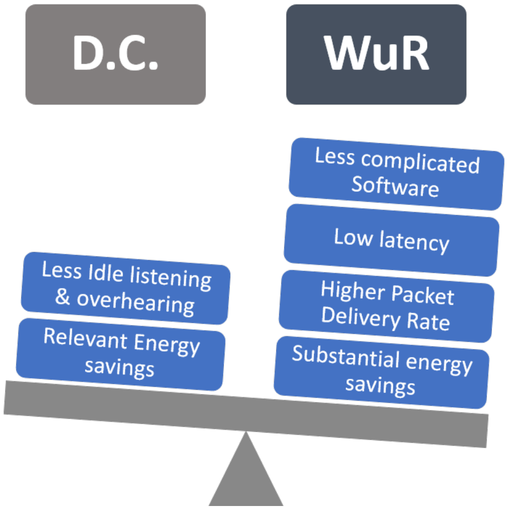


Fig. 2.5.: A qualitative comparison between Duty-cycling (D.C.) and Wake-up Radio (WuR) approaches for energy management in WSNs.

WuRx presents low receiving sensitivity (e.g., -55dBm, 50m, 10 kbps [17]), due to the passive demodulation of the carrier signal [77] and the low power consumption in WuRx designs–operating in the micro-Watts order [62], even in the order of nano-Watts [78, 79] that can be supplied using energy-harvesting techniques (e.g., radio-triggered [80]), not requiring any energy consumption from the SNs [74].

A limited receiving sensitivity also causes a higher power transmission on WuTx to reach the receivers at longer communication ranges. This condition implies an extra energy consumption [17]. Most research efforts are concentrated to achieve WuRx designs with higher receiver sensitivity, allowing long-range and low power transmission on WuTx [5].

During the WuR system operation, each sensor node needs to transmit a wake-up call signal before each data transmission. This condition results in overheads. Consequently, the energy consumption increases on the transmitter side, as well as, the traffic load over the shared communication channel. Some research efforts were proposed to reduce the wake-up overhead, such as the double modulation scheme in [81], where the sensed data to be transmitted is encapsulated into the payload of a wake-up call signal.

2.5 Remarks

In the literature, it is usually common to find WSNs designs that consider only one data-reporting scheme, neglecting the behavior of most monitoring applications (e.g., fire event monitoring), where both event-driven and time-driven modes are required simultaneously [3]. Hence, we did a systematic literature review based on the initial research question:

- *How can we integrate both time-driven and event-driven data-reporting modes to enhance the lifetime of M2WSNs while maintaining a desired event reporting accuracy?*

We presented a taxonomy of M2WSNs techniques with the most relevant MultiModal Switching and Network Management techniques, focused on D.C. and WuR approaches–those implemented in real motes. Finally, we found that most M2SMs presented in the literature were evaluated under simulated scenarios, without considering real hardware and environmental constraints that are essential for rigorous energy consumption and latency studies.

Some of the main challenges and future research directions in M2WSNs techniques from an energy-efficiency perspective are presented below.

MultiModal Switching Techniques. Most of the M2SMs (Table 2.3) tried to enhance the energy-efficiency of WSNs through a switching mechanism between EDR and CMnt, assuming an always-on operation. Indeed, we got a better performance when compared to applications that use only one data-reporting mode [6], but these techniques still present significant power consumption ($57.4mW$, see Section 5.3.2) that can be unsuitable for ultra-low-power applications, such as those required in hostile environments. Therefore, by combining M2SMs with D.C. techniques, we can achieve substantial energy-savings ($1.7mW$, see Section 5.3.2). However, we got a good energy performance but sacrificing the response time of the network upon the occurrence of an event. The main reason is that the transceiver of the SNs remains most of the time off, and given that during the active operation, there are only short periods when the SNs are listening to the channel.

Consequently, the sender node must wait until the receiver wakes up to transmit its data packet. Hence, higher data latencies are generated–due to the waiting time and overhearing issues. By implementing a WuR mechanism, such as the one presented in [82], the event-time response or end-to-end delay might be substantially reduced

Network Management Techniques. We considered two relevant approaches for energy-efficient M2WSNs, within the Network Management techniques: D.C. and WuR. D.C. is still an evolving research area. Most of the WSNs designers seek for very low duty cycles (below 1%) but at the cost of compromising the network performance (e.g., end-to-end latency) [18]. From energy conservation, D.C. is an essential technique for CMnt applications but suffers from latency issues (due to idle listening and overhearing issues). Hence, WuR approach is a promising technique for D.C. limitations [5, 17].

Currently, WuR approach (hardware and software) is in a developmental stage and presents excellent opportunities for research themes, as claimed by Oller *et al.* in [62]: “We strongly consider the on-demand nature and the energy savings provided by WuR as a decisive factor for rethinking applications from using traditional MAC protocols".

Djiroun and Djenouri in [5] confirmed that the use of a separate WuR eliminates all the drawbacks presented in D.C. approach, considering today as a good option in power-aware protocol design. However, the state-of-the-art in WuR hardware present low receiver sensitivity (e.g., -55dBm) thereby very short communication ranges (e.g., up to 40m) that could be insufficient for most large-scale monitoring applications. Though, the authors in [5] asserted that duty-cycling the WuR guarantee a larger wake-up range, similar to the communication range achieved with single radios. We increase the wake-up range, but at the cost of increasing the end-to-end data latency that is the headache of traditional MAC protocols [83].

Therefore, we came across with the following questions:

- *Should we consider a WuR or a D.C. approach or a combination of both approaches for energy management in monitoring applications using M2WSNs?*
- *Can a WuR approach prove more beneficial than a D.C. approach in the near future of real M2WSNs implementations?*

A promising solution, according to [5], is the design of effective asynchronous wake-up MAC protocols using path reservation wake-up techniques (i.e., data forwarding and wake-up messages transmission are performed simultaneously, such as in CTP-WUR mechanism [29]). A complete qualitative analysis on path reservation wake-up MAC protocols was developed in [5].

Finally, to reinforce the above idea, new asynchronous and more responsive to channel WuR MAC protocols are required for monitoring applications using M2WSNs, considering the dual radio setup of most WuR architectures. The protocol design might be largely simplified using always-on WuRx. Additionally, according to Piyare *et al.* in [17], up to now there is not a unified system and networking architecture for WuR approaches, where applications can be implemented, without relying on simulation tools, but on real implementations or testbed.

A Framework for M2WSNs

3

In this chapter, we introduce a framework for M2WSNs suitable for monitoring oriented applications with low bandwidth requirements. The framework follows a modular or layered approach, where each layer aims to fulfill specific tasks based on its own information, the functions provided by its adjacent layers, and the information resulting from the cross-layer interactions. Some techniques associated with each layer are described here and some others will be detailed in Chapter 5 and Chapter 6.

3.1 Overview

We propose a framework for M2WSNs appropriate for monitoring oriented applications that operates under normal circumstances and emergencies, using a dual-radio architecture based on the wake-up radio paradigm. Figure 3.1 shows an overview of the proposed framework.

The left-hand side of Figure 3.1 provides different modules that include the framework. From a traditional WSNs architecture perspective, the top module represents the application layer, and the bottom module, the physical layer. The intermediate modules, the network layer, and link layer, respectively. Each module aims to fulfill specific tasks. Starting from the bottom to the top, radio transceiver managing, medium access control and radio duty cycling, path-selection (routing) and packet-forwarding, data gathering, node scheduling and switching between different reporting modes. Besides, we introduce a cross-layer entity, whose primary function is to manage the information provided by the different modules, to operate efficiently, working as a whole system, not as independent layers.

A "strict interaction" refers to the interaction between two adjacent layers (e.g., application layer and network layer). While, a "cross-layer interaction" refers to the interaction between two o more layers that are not adjacent to each other via the cross-layer manager (CLM) entity, e.g., application layer and link layer. During one interaction, the layers exchange packet data units (information) required to execute a particular technique. The acronyms on the Figure 3.1 are presented in Table 3.1. i_m represents the data structure exchanged between the CLM and a particular layer.

In general, the proposed framework has the following features:

- **Responsiveness**. In emergencies, events and their associated data are propagated immediately towards the sink, throughout the network [84]. The M2WSNs do their

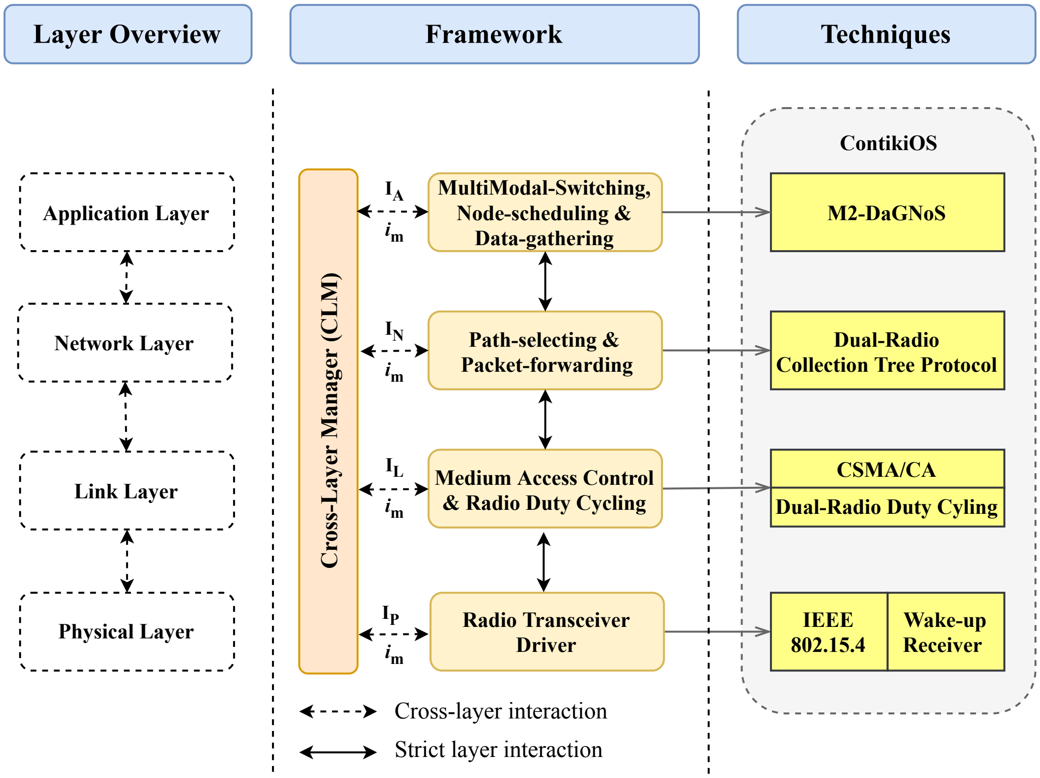


Fig. 3.1.: A general framework for M2WSNs: An overview and its associated techniques.

best-effort to report the first event packet to an observer (e.g., sink node), as soon as possible, managing the trade-off of latency for energy-efficient operations. The proposed framework provides this feature mainly by the implementation of a multimodal switching mechanism and a wake-up protocol based on the wake-up radio paradigm.

- **Energy-efficiency.** From a radio perspective, energy consumption is minimized to extend the lifetime of the network while providing sufficient data granularity to the sink. Primary, the application, link (radio duty cycling), and physical modules of the framework are oriented to manage this feature.

- **Distributed.** Sensor nodes within the M2WSNs can take actions autonomously regarding data gathering and reporting duties, without relying on sink's or neighbor's mediation, i.e., each sensor node runs the proposed techniques locally using its own collected data for decision-making and resource utilization. The framework performs this feature principally at the application layer.

- **Cross-layer interactions.** The framework allows interactions between non-adjacent layers within a sensor node's protocol stack. We chose a cross-layer approach to design the framework because it is considered more energy-efficient, scalable and allows a better distributed design than a traditional approach (e.g., OSI model) [85].

Tab. 3.1.: Cross-layer information types and contents.

Information type	Contents
Application (I_A)	(A_1) Switching status, (A_2) Node-queue[a], (A_3) Required Power Tx
Network (I_N)	(N_1) Sending status, (N_2) Neighbor table
Link (I_L)	(L_1) Wake-up period, (L_2) Retransmission status
Physical (I_P)	(P_1) Current Power Tx

[a] It refers to a set of id-node stored into a queue.

- **Low bandwidth.** The framework is designed for applications with low bandwidth, i.e., small data packet (without using a compression technique) and low transfer data rate requirements, less than or equal to 250kbps supported by the IEEE 802.15.4 standard [86], enough for monitoring of physical variables. For instance, the temperature usually requires a few bytes to be transmitted. For other types of data, such as video or sound, it may require a higher bandwidth (higher data rate) to be transmitted that is not within the scope of this dissertation.

The primary constraints to the proposed framework are related to mote limitations, i.e., motes are limited in memory, computation capacity, and amount of energy resources. Therefore, the framework is based on lightweight and power-aware techniques that are described in the subsequent sections.

3.2 Techniques

The right-hand side of Figure 3.1 presents the techniques used to address the requirements, the middle diagram, requested by the framework.

3.2.1 MultiModal-switching, Node-scheduling and Data-gathering

The top module or application layer of the framework is in charge of performing the switching between CMnt and EDR data-gathering modes based on the circumstances presented within the area of supervising, i.e., under an emergency, the sensor nodes employ the EDR capabilities, and during the development of an event and its period of calm (i.e., under a normal situation), the sensor nodes switch to the CMnt capabilities to periodically transmit up-to-date data of the event status towards the sink. This procedure is performed with the help of an event-driven node wake-up method combined with a parameter-based event detection (PED) algorithm. For further energy savings, a cyclical node sleep scheduling method is included. Hence, the M2WSNs can achieve better energy-efficient operations and best-effort in reporting accuracy for monitoring-oriented applications.

We propose the energy-efficient and distributed M2-DaGNoS technique that is a MultiModal mechanism for Data Gathering and Node Scheduling in M2WSNs monitoring applications. From the radio perspective, the M2-DaGNoS approach helps to minimize the energy consumption by managing a data reporting duty-cycling at the application layer combined with the radio duty-cycling at the link layer. The M2-DaGNoS mechanism, in comparison with most of those reported in the literature, employs a variable frequency for data reporting according to the circumstances, where the data reported is the average of several measurements of the physical variable during a time window. Besides, an implicit network time synchronization is added for data sensing and communications. Also, a radio-duty-cycling scheme and power-aware network mechanisms are combined with the M2-DaGNoS for further energy savings during the whole life of the nodes.

The M2-DaGNoS allows adjusting the power transmission to extend even more the lifetime of the SNs, via cross-layer interactions, i.e., during the network initialization, the maximum power is used, giving that it is assumed that all SNs within the same grid can communicate to build their node-queue. Afterward, the power transmission is reduced to a value sufficient to permit SNs to communicate with their nearest and best neighbor (i.e., parent node). Finally, it includes an enhanced version of the cyclical node sleep scheduling mechanism proposed in [25], combined with the PED algorithm described in [6], to manage the goals of both CMnt and EDR schemes.

The M2-DaGNoS mechanism can operate under single-radio and dual-radio architectures. The former employs only one radio with IEEE 802.15.4 support. The latter consists of two radio modules, one main radio for data transmission and reception procedures under the IEEE 802.15.4, and a secondary radio for wake-up signals reception at the same frequency operation of the primary radio (e.g., 2.4GHz). More details about the design, implementation, and performance evaluation of M2-DaGNoS is provided in Chapter 5.

3.2.2 Path-selection and Packet-forwarding

The path-selection and packet-forwarding tasks are essential to pull the data out of the network, i.e., to send data through a known path towards an observer (e.g., sink node). The network layer usually performs these tasks with the cooperation of the logical link control communication services (e.g., reliable unicast communication).

The framework allows sensor nodes within the M2WSNs to be self-organize in a tree-like topology, having the sink node as the root. The tree is dynamically created and maintained over time based on a route metric (rtmetric) that is a function of the expected number of transmissions (ETX) to the sink. Each node determines its rtmetric value based on the rtmetric of its parent that is its best neighbor, i.e., the node that minimizes the ETX to the sink. The sink has a rtmetric value zero, the other nodes in the tree have a higher rtmetric value, depending on how far they are from the sink and the current link conditions. Therefore, data are always forwarded from any node via a multi-hop path with the fewest ETX to the sink. The procedure described before is provided by the well-known technique named as Collect Tree Protocol (CTP) [87].

The framework is implemented in ContikiOS, a lightweight and flexible operative system for low-power WSNs [88]. ContikiOS offers a ready to use lightweight layered communication stack, known as RIME [89]. Within the RIME stack, there is an implementation of the standard CTP [87]. We chose Contiki CTP for the path selection and packet forwarding tasks in M2WSNs that offers a highly reliable multi-hop data delivering technique via a tree-like topology [90].

Figure 3.2 shows a flow chart that summaries the CTP operation implemented in ContikiOS. The Contiki CTP combines several mechanisms to operate, such as routing (tree creation), neighbor discovery and management, link estimation, and duplicating packet filtering that are included in Figure 3.2. Some procedures are not shown for simplicity such as timers expired procedures associated with the periodical report of announcement packets to populate the neighbor table, and to remove older neighbors to flush the neighbor table (memory). Finally, the update of the rtmetric (ETX) value is done with the equation (3.1) shown below, every time an incoming event occurs such as ACK or announcement packet arrival, where n and N are the entry index and neighbor table, respectively. More details of this technique are provided in [90].

$$rtmetric = \arg\min_{n \in N} \{rtmetric_n + ETX_n\} \tag{3.1}$$

The current version of CTP is implemented to support a single-radio architecture within a multi-hop communication. Hence, we have adapted the CTP for a dual-radio (2R) architecture, 2R-CTP–those processes highlighted in Figure 3.2. After the first data packet is transmitted to the next hop within the known path, the source node waits for an acknowledgment (ACK). In a dual-radio architecture based on a WuRx scheme, the first packet (i.e., WuS) is usually lost, giving that it is used for waking up the next hop in the path. Hence, the source should retransmit the data packet. Figure 3.3 shows the process of packet retransmissions from the source node perspective and the time awaking windows (T_{AW}) of the next hop in the multi-hop path (the gray dotted block). The interrupt signal represents the process of awaking the next hop when a WuS is received by its WuRx hardware. The colors of the arrows represent the different timers set in the source to retransmit a packet.

Hence, as presented in the Algorithm 1, between the first packet transmission and the second one (i.e., first retransmission), there is a period of $T_{AW}/2$, half the time of the period that the next-hop remains on to receive the next packet(s). If the second transmission fails, due

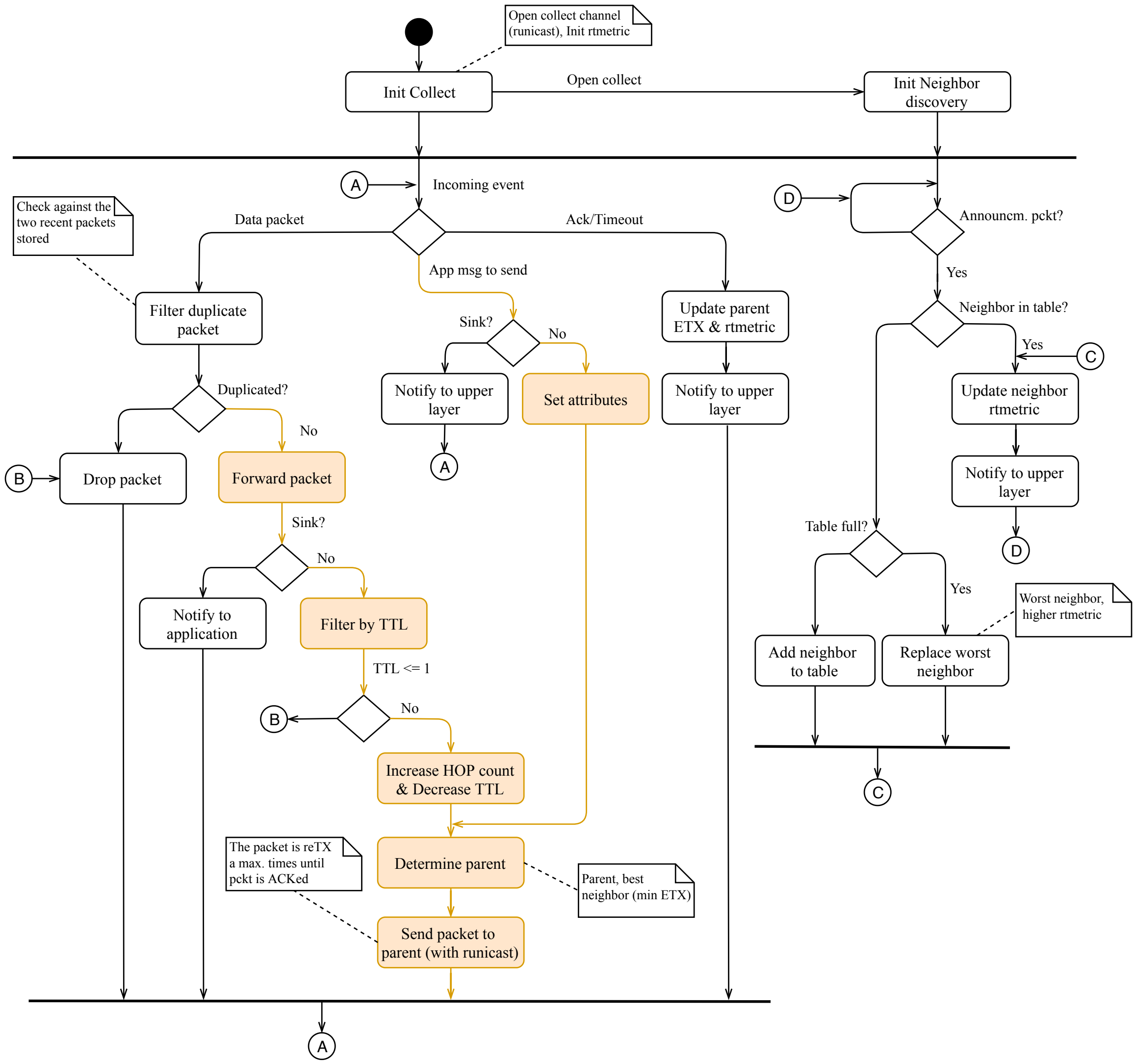


Fig. 3.2.: The collect tree protocol operation implemented in ContikiOS (Adapted from [90]).

to collisions or ACK loss, the data packet is retransmitted every $T_{AW}/4$, as shown in Figure 3.3, until the maximum number of retransmissions is achieved or the ACK is received. In this way, we can increase the probability of hitting the next-hop on its awaking period to receive the data packet and forward it to other hops in the path.

3.2.3 Medium Access Control and Radio Duty Cycling

Medium Access Control and Radio Duty Cycling (RDC) techniques have been fundamental in WSNs to operate efficiently and for energy saving purposes. The former is responsible for coordinating the channel access when sensor nodes need to transmit packets. The latter is in charge of controlling the sleep period of the nodes, i.e., to make sure that the node

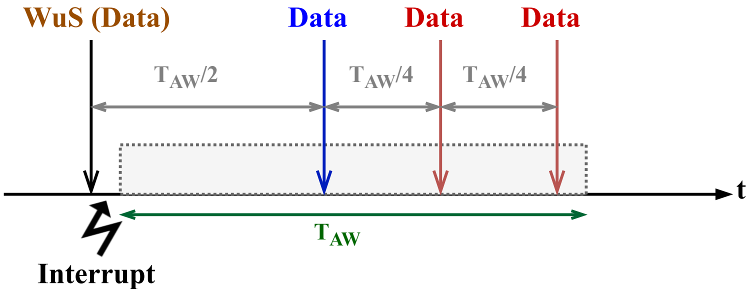


Fig. 3.3.: Data packet retransmission time adjustment for a dual-radio scheme using CTP.

Algorithm 1 2R-CTP: Data packet retransmission time adjustment

```
Input: $T_{AW}$, time awaking window; $rTX$, number of retransmissions; $rTX_{max}$, maximum rTX.
Output: $t_{rTX}$, data packet retransmission time.
procedure RTX-ADJUSTING($T_{AW}$, $rTX$, $rTX_{max}$)
    First packet sent                                        ▷ Usually lost (WuS)
    while No ACK received or $rTX! = rTX_{max}$ do
        if $rTX == 0$ then                                   ▷ 1st rTX
            Set $t_{rTX} = T_{AW}/2$
        else
            if $rTX > 1$ then                                ▷ Remaining rTX fail
                Set $t_{rTX} = T_{AW}/4$
                                        ▷ Increase possibilities of hitting next-hop
```

is awake to receive an incoming packet and to decide when is appropriated to transmit a packet. These techniques are usually implemented at the link layer.

Regarding the MAC technique, we employed the well-known *Carrier-Sense Multiple Access with Collision Avoidance* (CSMA/CA) method supported by the standard IEEE 802.15.4 [86] and ContikiOS. CSMA/CA is a widely used contention-based MAC protocol for wireless communications that do not require precise synchronization and is more adaptable to dynamic traffics. However, this protocol is collision avoidance, i.e., when a node is transmitting it can not detect any other transmission in the network; consequently, CSMA/CA is prone to collisions that imply additional packet re-transmissions at cost of extra energy consumption.

The CSMA/CA technique available in ContikiOS works as follows: Any packet received from the higher layer is enqueued in a FIFO manner in the link layer. Before transmitting the packet, the link layer delays the carrier sensing mechanism by $\alpha = 1/CCI$ seconds, where CCI is the channel check interval defined by the RDC technique (e.g., 8Hz). Afterward, it performs carrier sensing. If no carrier is detected, the packet is transmitted. Then, the link layer waits to receive an acknowledge (ACK) packet. Hence, a timer is set to a predefined interval (e.g., 192 μs [91]). If during this interval the ACK is successfully received, the node backs off and then it sends the next available packet on the queue. Otherwise, if the timer expired and no ACK is received, the node retransmits the packet provided that it does not exceed the maximum number of re-transmissions (e.g., two attempts) or else, the packet is dropped, and the transmission fails. If the channel is busy, the node backs off for some time at least α seconds, and then, attempts again to transmit the packet. In case of collisions, before attempting to retransmit the packet, the node backs off a random time which is in the interval $[\alpha, \alpha + \alpha * 2^{BE}]$, where BE is the back-off exponent, i.e., the number of allowed transmission attempts of the packet [91].

A dual-radio duty-cycling technique, 2R-MAC, is proposed to manage the switching on and off procedure of the radio transceiver, i.e., the radio should keep off as much as possible to achieve better energy-saving results, but at the same time, the radio should be able to receive an incoming packet. The 2R-MAC technique is an RDC protocol that works under a dual-radio architecture based on WuRx. It allows the coordination between the main radio and the WuRx and offers an interface to the higher layers protocols as CSMA/CA and 2R-CTP via cross-layer interactions.

The 2R-MAC technique is a modified version of the W-MAC protocol proposed in [82]. The W-MAC has been developed for the sky-mote (CC2420) as an emulator plugging for Cooja. W-MAC operates on in-band channels (at 2.4GHz) and ID-based schemes (i.e., with addressing support), and it is compatible with the CSMA and RIME stack. Therefore, multi-hop communications are supported. Besides, 2R-MAC operates similar to W-MAC but under a broadcast-based wake-up scheme, i.e., non-addressing, implemented based on [92]. More details about the design, implementation, and performance evaluation of 2R-MAC are provided in Chapter 6.

3.2.4 Radio Transceiver Driver

The radio transceiver driver is responsible for controlling the physical radio, such as the TI CC2420 or TI CC2520, that are a low-cost and low-power single-chip IEEE 802.15.4 RF transceiver for the 2.4 GHz unlicensed ISM band [86]. The driver is implemented in software and provides essential services to higher layers (e.g., for RDC): switching the transceiver on and off, sending and receiving packets, checking for channel availability, and setting configuration parameters (e.g., power transmission setting, channel switching, addresses node changing) [93]. Besides, the driver works with the framer 802.15.4 available in ContikiOS for parsing and generating of formatted packets with the IEEE 802.15.4 frame complaint.

For data transmission and reception, the framework is designed to support an IEEE 802.15.4 radio complaint driver that is available in ContikiOS, such as the traditional cc2420 and cc2520 drivers. For wake-up signal receiving, i.e., the detection of a radio frequency data signal, the framework provides a WuRx driver for in-band channel and broadcast-based wake-up scheme, i.e., non-addressing support. It worth mentioning that the wake-up radio paradigm is started to be employed in WSNs as a solution for the low-duty-cycling mechanisms that usually waste energy due to idle listening, and increase the reporting latency due to the long-sleep intervals [94].

The WuRx driver is design to receive any data signal with a carrier frequency of 2.4GHz and then, to trigger a digital signal, via a general port input/output pin (GPIO) from the MCU available in the WuRx hardware to the data line that interface with the CPU of the main radio (i.e., the data radio), to wake-up the mote and receive the data packet. The framework does not support a wake-up transmitter driver since both the data radio and the wake-up receiver radio operate at the same frequency. Hence, the wake-up signal can be emitted by the main radio to wake-up the next hop in the routing path.

3.3 Cross-Layer Manager

The framework follows a cross-layer design approach [85] that preserves the traditional WSNs layers, i.e., from application to physical, but interconnects them through a cross-layer manager entity, working as a whole system, not as independent layers, making the framework more energy-efficient, responsive and distributed when compared to a traditional approach, such as the OSI model.

The *Cross-Layer Manager* (CLM) block in Figure 3.1 serves as an interface between the higher layers and the physical layer. Figure 3.1 shows only the cross-layer interactions associated with the main node, i.e., the WuRx is an independent module connected to the main node via GPIO. Hence, the WuRx module does not have direct participation in the cross-layer interaction. The CLM block manages the information shared by the different modules to operate reliably and efficiently regarding energy savings and reporting accuracy. We define four types of information: Application (I_A), Network (I_N), Link (I_L), and Physical (I_P) that are transferred by its corresponding layer to the CLM. Then, the CLM provides this information by request via a data structure, i_m. The information is stored in local data structures of the motes. Table 3.1 shows the information shared with the CLM.

Figure 3.4 shows the cross-layer interaction between the modules of the framework that are managed by the CLM. The link layer shares with the network layer the wake-up period (i_{L_1}) selected or configured by the RDC for packet re-transmissions purposes. Besides, the re-transmissions status (i_{L_2}) is provided by the link layer for sending timeout adjustments. The network layer shares with the application layer the sending status (i_{N_1}) (busy or free), for messages management, and the neighbor table (i_{N_2}), for the node queue building and maintenance procedures. The application layer shares the switching status (i_{A_1}) (EDR, CMnt)

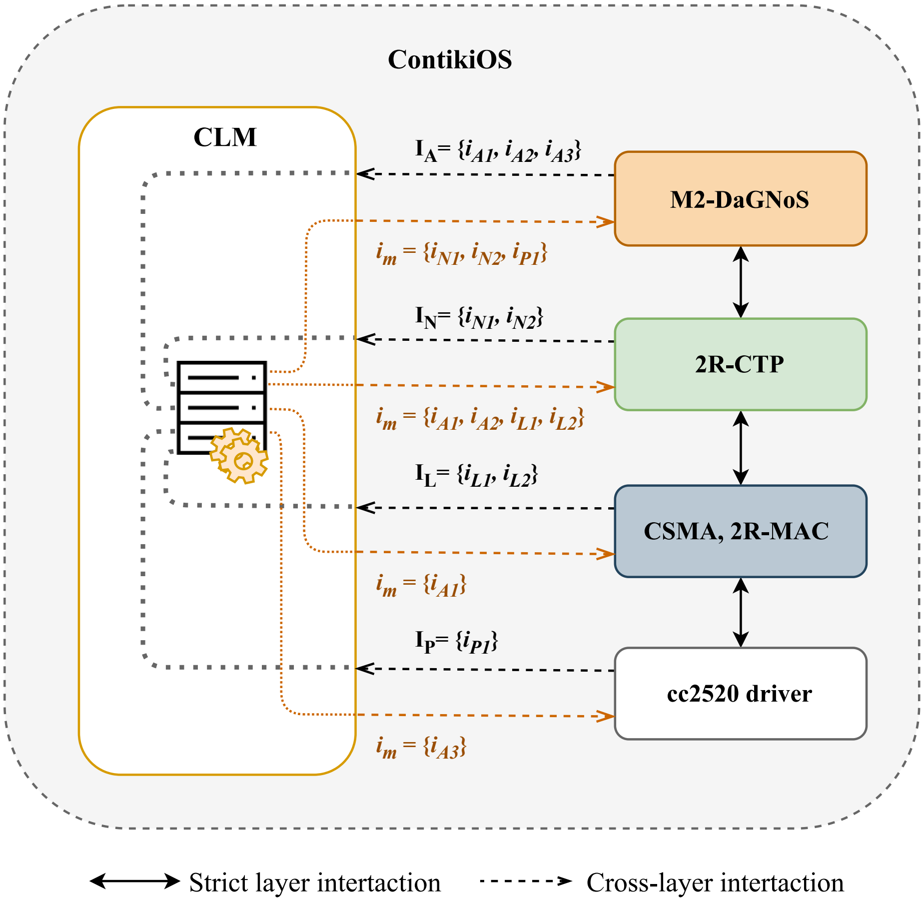


Fig. 3.4.: Cross-layer interactions between modules.

with the link layer, for radio duty cycling selection and adjustment; with the network layer, the switching status (i_{A_1}) and node queue information (i_{A_2}, generated by the M2-DaGNos technique) for priority packet management during emergencies, and dead node management, respectively. Besides, the application layer shares with the physical layer the desired power transmission (i_{A_3}) for energy reduction during normal operation after finishing the node-queue construction. The physical layer provides the current power transmission (i_{P_1}) to the application layer for verification purpose.

We are aware of that other interactions may exist in a cross-layer design approach such as between link to application layers, and network to link layers, but for the proposed framework operation, there is no additional information that might be considered. However, it could be analyzed in future work regarding the trade-off latency-energy consumption.

3.4 Remarks

In summary, this chapter provides an overview of the main features and techniques of the proposed framework for M2WSNs suitable for monitoring oriented applications with low bandwidth requirements that operate under normal circumstances and emergencies following a dual-radio architecture and cross-layer approaches. The framework design follows a modular approach, where each module cooperates to manage the trade-off of latency for energy-efficient operations in M2WSNs. Besides, the framework includes a cross-layer entity, whose primary function is to manage the information provided by the different modules, to operate efficiently, working as a whole system, not as independent modules as done by traditional network approaches such the OSI model.

An Energy Consumption Model based on Wake-up Schemes

4

Energy consumption is one of the most significant concerns in WSNs [5]. Traditionally, sophisticate power-aware wake-up techniques have been employed to achieve energy efficiency in WSNs, such as LDC protocols using a single radio architecture. These protocols achieve good results regarding energy savings, but they suffer from idle-listening and overhearing issues, that make them not reliable for most ultra-low power demanding applications [62]. Currently, WuRx based protocols, under a dual-radio architecture and always-on operation, are emerging as a solution to overcome these issues, promising lower energy consumption when compared to classic wake-up protocols, especially, in multi-hop communications [83]. In this chapter, we present an energy consumption estimation model that considers the behavior and performance of wake-up protocols based on WuRx and the traditional LDC schemes employed in M2WSNs. This model allows us to get more insights into the differences between both approaches and for considering a multimodality feature in WSNs by combining different transceivers and reporting protocols from an energy-efficiency perspective.

4.1 Problem Description

Typically, WSNs are composed of a large number of tiny SNs that are commonly battery powered and have limited energy resources. To save energy and to extend the lifetime of the SNs to several years, sophisticated power-saving techniques must be deployed [5]. Thereby, a considerable amount of energy can be saved, while a high monitoring quality is maintained if the behavior of the SNs can be adapted dynamically to the current conditions of the system. For example, if no activity takes place in the environment, the SNs could go into sleep mode. If only spare events take place, then SNs may run an LDC protocol, while in phases of high and critical activity in the surrounding, the SNs should be very reactive to forward data with very low latency. Such context-aware adaptivity can be achieved in several ways in which the transceivers and the transmission protocols play a crucial role in implementing this adaptivity and conserving energy [17].

M2WSNs allow using different types of transceivers and running M2SMs between different protocols for reporting, depending on the current context of the environment. For instance, in some regions with high activity, the SNs start to generate sensed data that must be transmitted to the sink. Other SNs, outside of these regions, are not aware of the high activity, and they become aware of this situation by the high amount of transmissions they have to forward to the sink or by receiving an assisting message transmitted by a working node in the region.

Traditional MAC protocols for WSNs are based on LDC approaches, where SNs switch between sleep and active states, and vice versa, following a predefined or on-demand scheduling [22]. LDC approaches help reducing the energy consumed by overhearing and idle listening. However, this reduction is insufficient for low-power demand WSNS designs [62], where SNs are required to save energy as much as possible to extend the network's lifetime. Therefore, a WuRx approach has been proposed recently to overcome LDC limitations [5].

A WuRx is an ultra-low-power radio hardware that is commonly added to SNs as a second radio, as shown in Figure 2.3. Some WuR prototypes are detailed in [17]. Typically, a WuRx is periodically listening to the channel for a pre-defined WuS that activates through an interrupt signal, other electronic parts of the SNs, for instance, the main micro-controller. The wake-up receiver can be listening for WuS in the same frequency band or at different frequencies of the main radio transceiver. The former is know as *in-band channel*, and the latter, *out-of-band channel* [17]. The *in-band channel* is cheaper because we can use the same main radio transceiver to transmit the WuS but at the cost of dealing with interference within the neighborhood that works at the same band. The *out-of-band channel* approach may decrease the interference issue, but it usually adds complexity and extra cost to the system design (two radio, one for WuS transmission, and another for data communication). However, compared to "classical" low-power radios, the power consumption of a WuRx is several orders of magnitude lower than these radios, allowing to keep it always-on [17]. Consequently, a WuRx can eliminate the idle-listening and waiting time issues, and help reduce the energy consumption and latency [5].

Some WuRx implementations have dedicated circuitry to perform an addressing mechanism by decoding a destination address contained in the packet header. Therefore, only the designate node is woken up instead of the entire neighborhood. This feature might allow solving the overhearing issue presented in LDC approaches. In this sense, there are mainly two manners to consider the recipient of a WuS. On the one hand, a source node can reach all the SNs within its neighborhood by a *broadcast-based wake-up*. All SNs within the source range receive the WuS. On the order hand, a source node intends to reach only one node within its range using dedicated circuitry. This scheme is usually known as *ID-based scheme* [17]. The latter is intended for selective wake-up addressing, where only the node with a specific ID is woken up. This scheme helps to reduce the false-wake-ups and the overall energy consumption of the whole M2WSNs (usually in large-scale deployments), but it requires a decoding process which is usually performed by an external micro-controller that adds an additional energy requirement to the WuRx power supply. Besides, the WuS

Tab. 4.1.: Comparison of relevant energy models based-on WuRx schemes for WSNs

Authors	Network Communication	Channels	WuRx Modeling	Error Approaching
Seyed *et al.* [95]	Single-hop	In-Band	Only addressing	Bit-error rate, wake-up miss and false alarm probabilities
Lont *et al.* [96]	Single-hop	Out-of-Band	Only addressing	Packet missed and false wake-up probabilities
Zhang *et al.* [97]	Single-hop	Out-of-Band	Only addressing	Miss detection and false alarm probabilities
Zhang *et al.* [98]	Single-hop	In-Band	Only addressing	None
This work	**Multi-hop**	In-Band	Both addressing, non-addressing	Error-free (no retransmissions)

packet needs extras bits (e.g., 2 bits [17]) for the destination address, that might require extra transmit time, hence extra energy consumption. The former can contribute to reduce the end-to-end latency, because the node does not decode the incoming WuS, but it might increase the false-wake-ups that may be potentially costly regarding energy consumption. Therefore, there is a trade-off to be made between latency and false-wake-ups reduction.

Piyare *et al.* [17] made a comprehensive overview of WuRx based on MAC protocols and offered an extended taxonomy of WuRx based on routing protocols compare to Djiroun and Djenouri's work [5]. They concluded that the lifetime of the SNs could be further extended by combining WuRx capabilities with selective addressing and routing duties (e.g., T-ROME [22]) while meeting latency requirements comparable to SNs that use a single radio.

In Table 4.1, a comparison is made between relevant energy consumption models proposed in the literature and our approach. Most of the work only applies for single-hop MAC protocols and two-channel solutions. Our model takes into account multi-hop communications and the energy budget for the whole network, and the main radio and WuRx use the same channel (in-band-solution), whereby only very few works exist in that area [17]. Finally, our model allows for modeling addressing and non-addressing WuRx when compared to other models that focus only on addressing.

In the following sections, we provide a parameterized energy estimation model that allows us to model the behavior of particular SNs and the whole M2WSNs. We consider that this model serves as a point of comparison between wake-up protocols based on WuRx and LDC schemes regarding the power consumption. We show that WuRx with addressing will not significantly save energy when compared to WuRx without addressing and that in some situations, an LDC scheme outperforms a WuRx scheme, while in some other situations, it is the other way around, giving a strong motivation for using multi-modal approaches in WSNs. Therefore, the main contribution of this chapter is summarized as follows: An energy

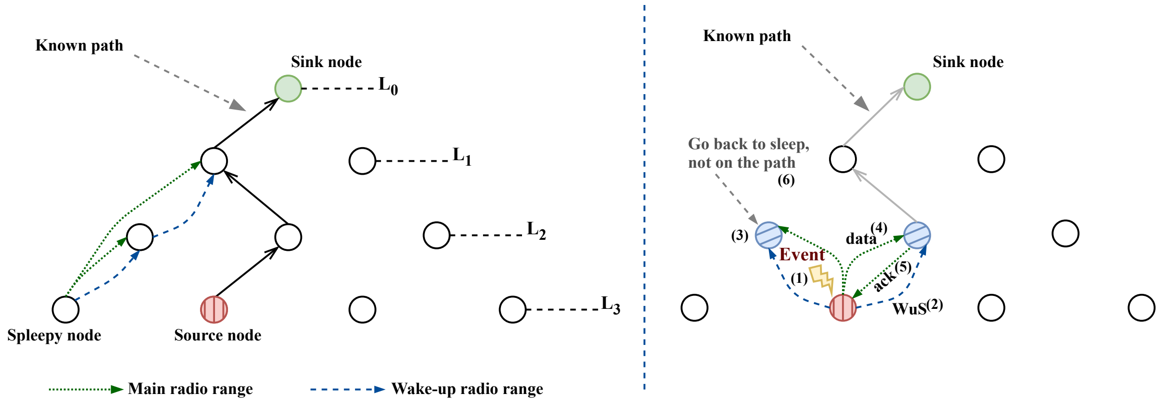


Fig. 4.1.: A dual radio multi-hop communication schematic on a tree-like topology (Based on [22]). The right sketch shows a multi-hop operation of the wake-up protocol proposed. The event data packet is propagated from the source to the sink through a known routing path previously defined.

consumption estimation model for M2WSNs that considers the behavior and performance of wake-up protocols, mainly, those based on WuRx.

4.2 Energy Consumption Models

Normally, the wake-up protocol is integrated into the link layer of a layered architecture for M2WSNs (Figure 3.1) that includes an application layer which runs M2SMs using the information provided by its adjacent layer. A network layer that performs topology management duties, e.g., path selection, and packet forwarding. A link layer that executes sleep-wakeup duties combined to medium access control with retransmission functions, aiming to minimize the energy consumption at the physical layer by reducing the transmission power while providing a high monitoring quality. In this chapter, we focus only on the lower-layers and consider a dual radio communication (main radio and WuRx) at the physical layer, both radios sharing the same channel (in-band solution).

Figure 4.1 shows an example of a multi-hop communications on a tree-like topology for reporting an event and its associated data to the sink using a dual radio (left scketch). The right sketch shows the process of transmitting one packet toward the sink based on a broadcast-based wake-up protocol: (1) A source node detects an event, reports and propagates it through a known and reliable multi-hop routing path towards the sink, previously defined by any "classical" routing protocol (e.g., CTP or RPL as done in [82]). (2) The wake-up protocol operates under a transmitter-initiator scheme, where the source starts the communication by first sending a WuS packet (the same event data packet) using its main radio. (3) The WuS packet wakes up all potential receivers, (i.e., child and parents) within the WuR range (i.e., those SNs that have a WuRx integrated). (4) After sending the WuS,

Tab. 4.2.: List of variables of the energy consumption model

Variable	Description
$P_{Set}^{\phi}, T_{Set}^{\phi}$	Power and time required to settle WuRx or switch between idle and Tx (transmission) or Rx (Reception) states, where $\phi \in \{wrx, mrx, mtx\}$
P_{Wake}, T_{Wake}	Power and time required for wake-up procedures
P_{Idle}, T_{Idle}	Power and time consumed in idle state
P_{Tx}, T_z	Power and time required for z packet Tx, where $z \in \{data, ack, WuS\}$
P_{Rx}, T_y	Power and time required for y packet Rx, where $y \in \{data, ack\}$
P_{DLPM}, T_{Sleep}	Power and time consumed in the deepest sleep state
P_{LPM}	Power consumption of SNs in low power mode (LPM) mode
P_{node}	Power consumption of SNs when transmitting, receiving, listening, switching between states.
$P_{\triangle DLPM}$	Power budget allocated only to the active mode excluding the power budget allocated to the deepest sleep mode.
P_{wrx}	Power consumption of WuRx during its operation
T_{sw}^{sleep}	Time required to switch between active and sleep states
N	Total number of SNs in the M2WSNs
h	Number of hops in the known path
$nbor$	Number of neighbors that are woken up
$mbte$	Mean time between two events
T	Total time duration of an active cycle
t_{run}	Total runtime of the M2WSNs

the source waits for a short time, and then emits an event data packet. (5) Afterward, the source waits for an acknowledge packet (ack). If no ack is received during a predefined time, the sender transmits the same data packet again, until an ack arrives or the number of retransmissions is exhausted. (6) The non-destination SNs turn on their main radio after receiving the WuS, remain active until a data packet is received, but then go back to sleep because the destination address does not match their address. Finally, this procedure is repeated in each hop within the multi-hop path until the sink receives the event data packet. In Table 4.2, we provide a list of variables and their corresponding description that are used in the diagrams and the later analysis.

The state machine diagram for such a dual radio setup of a single node is shown in Figure 4.2. The dashed rectangles are transition states, and the power consumption of a state, $P_{State}^{component}$, and the period a node remains in each state, $T_{State/packet-type}^{component}$, are shown in

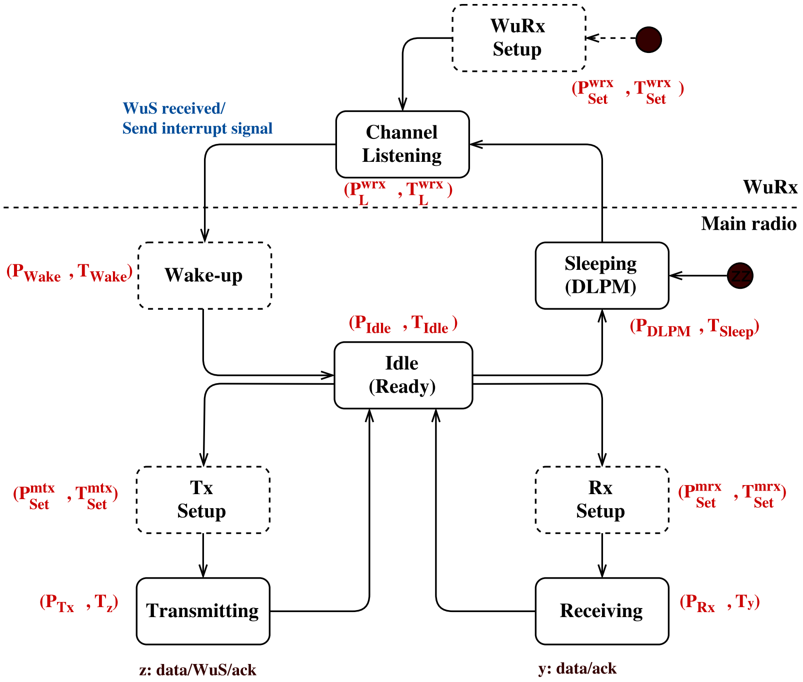


Fig. 4.2.: Dual radio state machine diagram (Based on [101, 96, 95]).

round brackets. The WuRx has one main state, $channel - listening$, and a transition state, $WuRx - Setup$, that models its initial setup. After the setting up state, the WuRx remains listening to the channel, waiting for a WuS to arrive. Upon a WuS, the node is woken up by an external interrupt signal generated from the WuRx. In this case, the main radio of the node stays in its deepest sleep mode (DLPM) with the lowest power consumption P_{DLPM}. Otherwise, if the node runs an LDC protocol without WuRx-support, some timers are needed to wake the node up periodically, i.e., it stays only in a low-power mode (LPM) with power consumption P_{LPM}, and typically, P_{LPM} is larger than P_{DLPM} [99, 100].

During the active mode, the main radio remains on its $Idle$ state, and switches between $Transmitting$ and $Receiving$ states depending on the task to perform: (1) to transmit a data or Ack packet to the next hop in the routing path or (2) to receive a data or Ack packet from the previous hop (to process it or to relay it). The transitions to set up the main radio are modeled in $Tx\ Setup$ and $Rx\ Setup$ states (e.g., data encapsulation), respectively. After performing these tasks, the main radio returns to its $Idle$ state, and then to the $Sleeping$ state, where it remains in the deepest low power mode.

In the subsequent sections, we present our underlying assumptions and briefly describe the protocol schemes for both cases, i.e., using the wake-up protocol based on the WuRx and based on the LDC scheme. Based on these descriptions, the energy model for delivering a packet over a single path to a sink is presented that resembles already known models. Finally, these models are extended to cover the energy consumption of the whole network, based on the node density and event rates.

4.2.1 Basic Assumptions

In our analytical model and to compare the benefits of a WuRx and an LDC configuration for M2WSNs, some underlying assumptions are made:

1. The WuRx and the main radio share both the same channel. As a consequence, transmissions of the main radio are detected by the WuRx. Therefore, each sensor node needs, besides the main radio, only a WuRx, and not a wake-up transmitter.
2. Both radios have the same communication range.
3. We do not care about the network topology and routing protocols. We assume that a sensor node knows the address of the next hop for delivering a data packet to the sink.
4. Transmitted data message is very small. Therefore, we assume that it is enclosed in the wake-up packet.
5. For the sake of simplicity, we assume that the power consumption (P_{node}) of the main radio is the same for the wake-up and idle periods, receiving and transmitting a packet, and switching between states.
6. For energy estimation, we assume no packet loss during communication, i.e., an error-free channel.
7. All SNs sleep in the deepest operating sleep mode, until they are woken up for their active period.

4.2.2 Wake-up Protocol based on a WuRx Scheme

Figure 4.3 presents the behavior of the proposed wake-up protocol using a WuRx scheme when delivering a data packet to the sink through a known routing path. The diagram shows the packets transmitted using the main radio.

The timing diagram in Figure 4.3 presents the interactions between the SNs within the established path to the sink. The wake-up protocol operates under a transmitter-initiator scheme, where the source (i.e., the sender node) or working node starts the communication by first sending a WuS packet using its main radio and then data packets with destination address. After sending a data packet, the sender waits for an acknowledge packet (ACK) (e.g., $T_{Idle} + T_{Set}^{mrx}$), and if no ack-packet is received, the sender transmits the same data packet again, until an ack-packet arrives or the number of trials is exhausted. The WuS packet wakes up all potential receivers, i.e., those SNs within the interference range of the sender node (as shown in Figure 4.1 step 3). Hence, also the non-destination SNs turn on their main radio, remain active until a data packet is received, but then go back to sleep

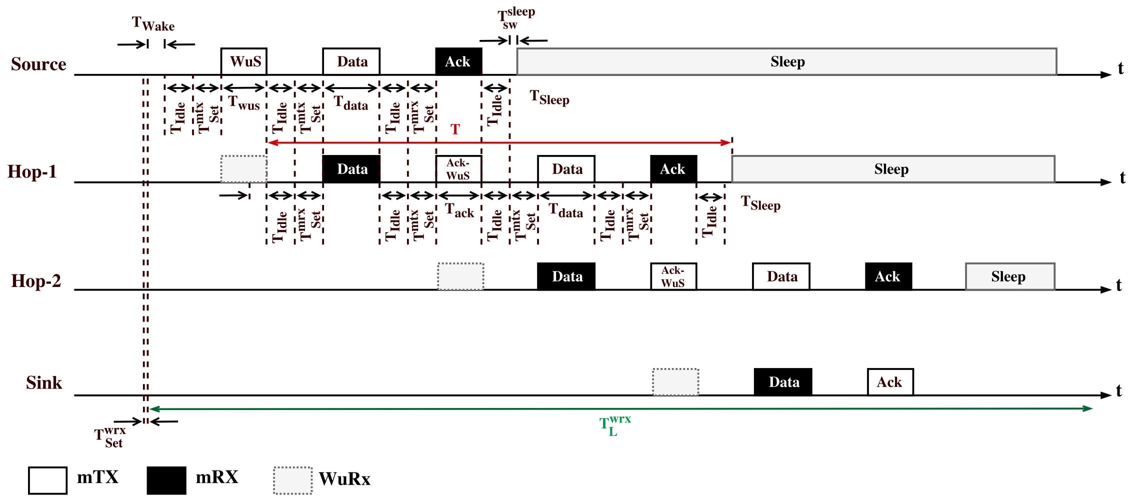


Fig. 4.3.: Timing diagram for delivering one event data packet over hops through the known path (Based on [96]) using a WuRx scheme. The non-destination SNs (not within the known path) interactions are not shown in the diagram. The white block (mTx) represents that the main radio is in the transmitting state, while, the black blocks (mRx), in the receiving state. The dotted gray rectangle (WuRx) indicates that the node uses its WuRx hardware to receive wake-up signals. The gray rectangle with solid lines refers to that the main radius remains in the sleeping state.

because the destination address does not match their own address (refer to step 6 in Figure 4.1). For simplicity in Figure 4.3, the non-destination SNs interactions are not shown.

The designated receiver (i.e., $Hop-1$, the sensor node within the multi-hop routing path) should receive one of the subsequent data packets, and then, send an ack-packet back to the sender. This ack-packet also serves as a WuS-packet that wakes up the next hop (i.e., $Hop-2$), and $Hop-1$ can forward the data packet to $Hop-2$, right after sending the ack packet. Finally, $Hop-1$ goes back to sleep. This procedure is repeated throughout the hops in the known path until the data packet is delivered to the sink, as shown in Figure 4.3.

Energy Model for a Single Path

Based on the assumptions made and the timing diagram of Figure 4.3, the energy budget (from the main radio perspective), E_{hop} in equation (4.1), allocated to a single intermediate node (e.g., $Hop-1$) on a known path is given by:

$$\begin{aligned} E_{hop} &= (T_{wake} + 2T_{data} + 2T_{ack}) \times P_{\Delta DLPM} \\ &+ (4T_{Set}^{\phi} + 5T_{Idle} + T_{sw}^{sleep}) \times P_{\Delta DLPM} \end{aligned} \tag{4.1}$$

where $P_{\Delta DLPM} = P_{node} - P_{DLPM}$ is the power consumed in active mode on top of the power consumed in DLPM.

Consequently, the energy budget allocated for transmitting a single data packet over h many hops on the known path is given by the equation (4.2).

$$e_{path} = (h-1) \times E_{hop} + E_{source-sink} \tag{4.2}$$

The first term gives the power consumption of the $h-1$ intermediate SNs on the path, while the second term complements the power consumption of the source and sink, $E_{source-sink}$ in equation (4.3) detailed below:

$$\begin{aligned} E_{source-sink} &= (2T_{Wake} + 2T_{data} + 3T_{ack}) \times P_{\Delta DLPM} \\ &+ (6T_{Set}^{\phi} + 7T_{Idle} + 2T_{sw}^{sleep}) \times P_{\Delta DLPM} \end{aligned} \tag{4.3}$$

Depending on the WuRx used (supporting addressing or non-addressing), also other SNs in the surrounding of the SNs on the path are woken up. In our model, we denote this number of nodes as $nbor$ and assume that in an M2WSNs with equally distributed SNs, $nbor$ is constant for every node. If the WuRx supports addressing, then $nbor = 1$, otherwise, we assume that each node on the path wakes up $nbor > 1$ SNs in its surrounding. These non-destination SNs stay awake until they have received a data packet with the destination address. Afterward, SNs realize that they are not the destination and can go back to sleep. Hence, the energy consumed by these SNs during their active period, $E_{non-dst}$, states in equation (4.4).

$$\begin{aligned} E_{non-dst} &= (T_{Wake} + T_{data}) \times P_{\Delta DLPM} \\ &+ (T_{Set}^{\phi} + 2T_{Idle} + T_{sw}^{sleep}) \times P_{\Delta DLPM} \end{aligned} \tag{4.4}$$

Therefore, the total energy consumed by each event to deliver it towards the sink (only if a wake-up event occurs), $e_{wakeup_{event}}$, considering equations (4.2) and (4.4), is given by the equation (4.5).

$$e_{wakeup_{event}} = e_{path} + (nbor - 1) \times h \times E_{non-dst} \tag{4.5}$$

Energy Model for the Complete M2WSNs

Now, we provide the model for the energy consumption of the whole M2WSNs. We divide the total energy consumption of the M2WSNs in two parts, as shown equation (4.6).

$$e_{total} = e_{base} + e_{active} \tag{4.6}$$

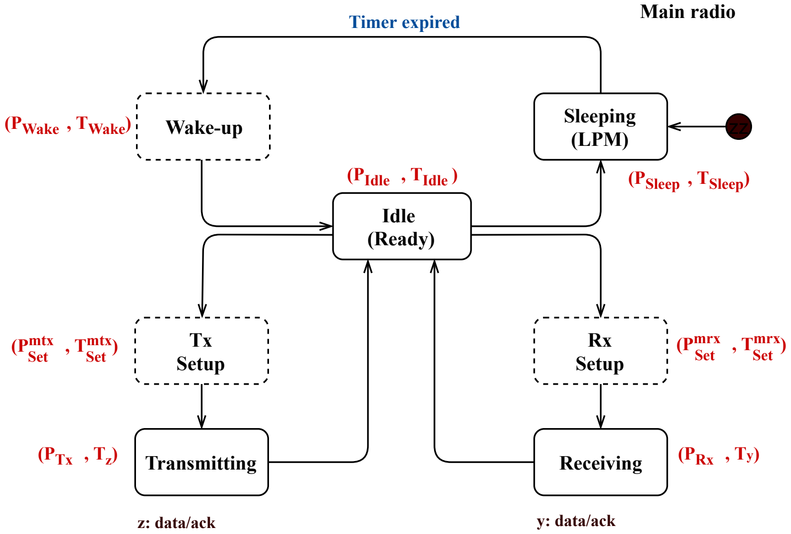


Fig. 4.4.: Single radio state machine diagram for LDC schemes.

The first term covers the base-energy consumption that is always present in DLPM. e_{base} is the power consumed in the DLPM by each node over time and its WuRx, during the total runtime of the M2WSNs, and it is defined as shown in equation (4.7).

$$e_{base} = N \times t_{run} \times (P_{wrx} + P_{DLPM}) \tag{4.7}$$

The second term in equation (4.6) covers the energy consumption on top of the base-energy-consumption during the active period of SNs. Depending on the activity in the network, the energy budget for all events is given by the equation (4.8).

$$e_{active} = (t_{run}/mtbe) \times e_{wakeup_{event}} \tag{4.8}$$

where $t_{run}/mtbe$ is the number of events during the total runtime (t_{run}).

4.2.3 Wake-up Protocol based on an LDC Scheme

The wake-up protocol presented in this section resembles the ideas of already existing protocols based on LDC schemes such as those proposed in [102]. The protocol follows the radio state machine introduced in Figure 4.2, without the states related to the WuRx, a transition between Sleeping and Wake-up states is added (that occurs after a timer expired), and use LPM instead of DLPM. Hence, the single radio state machine for LDC schemes results as shown in Figure 4.4.

The wake-up protocol operates under a sender-initiated LDC scheme. Every sensor node periodically wakes up to listen to the communication channel, and to determine if there are

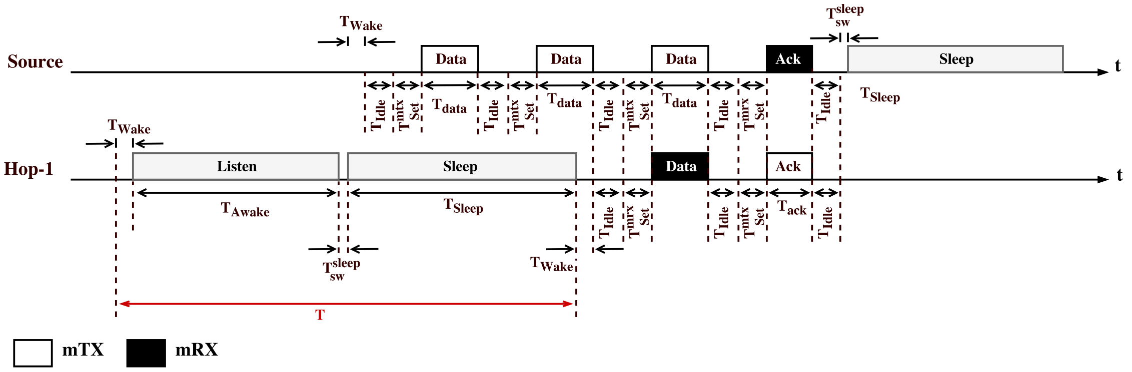


Fig. 4.5.: Timing diagram for delivering one event data packet over a single hop using an LDC scheme.

potential incoming data packets during a T_{Awake} period. If no data packet is detected, the node goes back to LPM and sleeps until its next scheduled wake-up interval (e.g., T_{Sleep}), as shown in Figure 4.5 (i.e., refer to $Hop-1$ time line).

When a sender node has a data packet to transmit, it repeatedly sends a beacon to its neighbors until an ack-packet is received (every $T_{Idle}+T_{Set}^{mtx}$). The beacon is the full data packet with the destination address (useful if a payload is small). Therefore, only the designated receiver (e.g., $Hop-1$) acknowledges the received data packet. After receiving the ack packet, the sender node stops transmitting the data packet and goes back to the sleep period, as shown in Figure 4.5. This procedure is followed by each hop until the data packet is delivered to the sink.

Energy Model for a Single Path

By following an approach similar to that of Section 4.2.2, we came up with the energy budget allocated for delivering an event data packet to the sink over h many hops as states equation (4.9).

$$e_{path} = e_{hop} \times h \tag{4.9}$$

where $e_{hop} = e_{recv} + e_{fwd}$ is the energy budget allocated to a sensor node to forward an event to the next hop. This energy budget is composed of two parts. The first part, e_{recv}, covers the energy for receiving the packet, and is given by the equation (4.10).

$$e_{recv} = T_{ack} \times P_{node} \tag{4.10}$$

Receiving takes place during the regular listen period of a node, whose energy budget is already covered by e_{base}. For that reason, no additional energy budget needs to be allocated, except the energy to receive the ack-packet.

The second part, e_{fwd}, gives the energy budget required to forward the packet that takes place usually during the regular sleep phase of receiver SNs. In the worst-case, the sender hits the active-period of the receiver after time $T_{sleep} + 2T_{data}$. In the best case, the first data packet of the sender hits the active-period of the receiver. We assume that the active period of the receiver is hit on average after half the worst-case time. Hence, e_{fwd} is given by the equation (4.11).

$$\begin{aligned} e_{fwd} = &\{(T_{Sleep} + 2T_{data})/2 + T_{ack}\} \times P_{node} \\ &+ (T_{Idle} + T^{\phi}_{Set}) \times P_{node} \end{aligned} \tag{4.11}$$

Energy Model for the Whole M2WSNs

The total energy consumed by the whole M2WSNs is also composed by two part as states in (4.6), but the formulas for e_{base} and e_{active} are given by equations (4.12) and (4.13).

$$e_{base} = T_{on} \times P_{node} + T_{off} \times P_{LPM} \tag{4.12}$$

$$e_{active} = (t_{run}/mtbe) \times e_{path} \tag{4.13}$$

where T_{on} and T_{off} are the sum of the active and sleep times, respectively, of all SNs, as shown in equation (4.14).

$$\begin{aligned} T_{on} &= N \times (t_{run}/T) \times (T_{Awake} + T_{Wake} + T^{sleep}_{sw}) \\ T_{off} &= N \times (t_{run}/T) \times T_{Sleep} \end{aligned} \tag{4.14}$$

where $T = T_{Wake} + T_{Awake} + T^{sleep}_{sw} + T_{Sleep}$ is the duration of a whole wakeup-cycle as shown in Figure 4.5.

The e_{base} in (4.12) summarizes the energy budget needed for the regular wake-up and sleep cycles, i.e., if no transmission occurs at all in the M2WSNs, then only e_{base} is consumed. Meanwhile, the e_{active} in (4.13) is the energy budget allocated to the transmission of a single event to the sink. This energy budget includes only the additional energy needed on top of e_{base} for transmitting the event to the sink.

4.3 Evaluation of the Theoretical Models

Based on the proposed energy model (4.6), we can estimate the total power consumption for the whole M2WSNs using both schemes under different WSNs configurations varying the number of SNs (N), event frequency ($mtbe$), number of hops in the known path (h), and the number of woken up neighbors along the path ($nbor$). Thus, we perform a parameter

Tab. 4.3.: Setting parameters for the energy consumption model (based on [103, 99, 100]).

Parameters	Value	Parameters	Value
Air data rate	100 kbit/s	N	200
T_{ack}, T_{Wake}	1.28 ms	h	5
T_{data}	2.56 ms	$nbor$	50
T_{Sleep}	10 sec	$mtbe$	60 sec
T_{Awake}	$4T_{data}$	t_{run}	1 year
T_{Set}^{ϕ}	1 ms	T_{sw}^{sleep}	5 μs
T_{Set}^{wrx}	negligible	T_{Idle}	799 μs
P_{node}	$33mA \times 3.3V$	P_{wrx}	150 μW
P_{DLPM}	$0.1\mu A \times 3.3V$	P_{LPM}	$6.6mA \times 3.3V$

Tab. 4.4.: Comparison of the total energy consumption of the M2WSNs under two low-power modes.

Scenario	Value [μW]	Ratio (WuRx/LDC)		
		$mtbe = 60s$ $T_{Sleep} = 10s$	$mtbe = 600s$ $T_{Sleep} = 10s$	$mtbe = 600s$ $T_{Sleep} = 25s$
1	$P_{wrx} = 1$ $P_{LPM} = 2.97$ $P_{DLPM} = 0.33$	0.05	0.02	0.03
2	$P_{wrx} = 150$ $P_{LPM} = 2.97$ $P_{DLPM} = 0.33$	0.46	1.00	1.38
3	$P_{wrx} = 150$ $P_{LPM} = 21.80 \times 10^3$ $P_{DLPM} = 21.80 \times 10^3$	0.99	1.00	1.00

sensitivity study for different configurations (i.e., the parameter settings for the model). Table 4.3 shows the values employed in the models that are based on a mote constructed with the commercial micro-controller MSP430 [99] and transceiver C1101 at 868 MHz [103], along with a WuRx. This WuRx allows for permanently listening while consuming $150\mu W$ [100].

First, we analyzed the impact on the energy consumption when considering different low-power modes supported by the SNs employing the WuRx, as shown in Table 4.4. For the first scenario, we considered the classical use case where $P_{wrx} + P_{DLPM} < P_{LPM}$, with $P_{DLPM} = 0.1\mu A \times 3.3.V$ nine times lower than $P_{LPM} = 9 \times P_{DLPM}$ (assuming that all peripherals can be turned off). For this scenario, varying $mtbe$ and T_{Sleep} shows that a significant energy budget could be saved when this WuRx protocol is used, i.e., the ratio between WuRx and LDC power consumption ranges from 0.02 to 0.05. In the second scenario, we used the real P_{wrx} value, where $P_{wurx} + P_{DLPM} > P_{LPM}$. Although the power consumption of the SNs using the WuRx is higher than without WuRx, using the

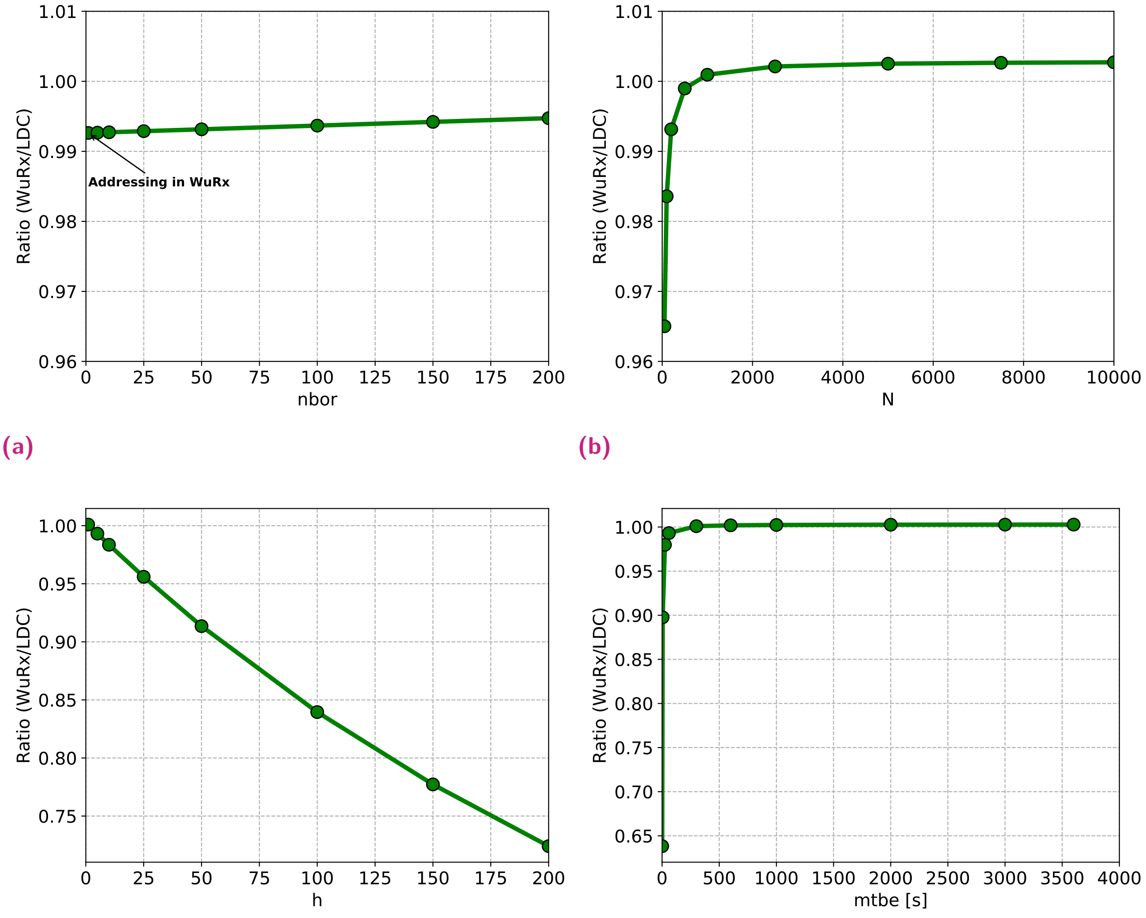
WuRx protocol still saves 50% of the energy budget. However, the energy consumption is very sensitive to the traffic load, as shown in Table 4.4, where the LDC scheme becomes better than WuRx scheme for long sleep periods and low event rates. Hence, in such situations, the LDC scheme should be used instead. In the third scenario, we assumed $P_{DLPM} = P_{LPM} = 6.6mA \times 3.3.V$, reflecting the case that the WuRx configuration cannot benefit on the SNs from a DLPM, e.g., due to some peripherals that must be turned on all the time. If we employ the same LPM, both wake-up protocols cause almost the same energy consumption, being WuRx scheme a little better than LDC scheme. This scenario is also less sensitive to the network activity (e.g., event frequency).

In sum, a dual radio communication becomes beneficial in scenarios of short and long sleep periods, and light and heavy event rates, when the energy consumption of the WuRx is in the order of some micro-watts, and the sensor node remains in its deepest low power mode (i.e., LPM4). For $P_{wurx} + P_{DLPM} \geq P_{LPM}$, the wake-up protocol based on WuRx performs better than LDC schemes in circumstances of high event occurrence rates, due to its always-on listening mode.

The results are obtained by varying the same single parameter in both models. For instance, in Figure 4.6a, the parameters $N = 200$, $h = 5$ and $mtbe = 60s$ remain fixed, while $nbor$ is changed between 0 to 200 to visualize the effect of $nbor$ over the total energy consumption ratio between both models. Besides, the study is made under the third scenario, where $mtbe = 60s$ and $T_{sleep} = 10s$. For each parameter value, the energy consumption in each model was computed, and the ratio, i.e., energy consumption WuRx-based model/energy consumption LDC-based model, is shown on the $y-axis$. Hence, if the ratio is less than one, the energy consumption in the WuRx-based model is smaller than the LDC-based model, and vice-versa.

The effect of varying the $nbor$ is negligible, according to the results presented in the top left graphic. When $nbor = 1$ (i.e., addressing in WuRx is used) the energy consumption is almost the same as for larger $nbor$ values (modeling no addressing). We can conclude that a broadcast-based wake-up is not energy costly when compared to an ID-based wake-up when the size of a sensor node's neighborhood increases and a high event rate traffic is considered. Besides, compared to a WuRx that supports a WuS decoding operation, the total energy consumed for waking up the microcontroller using an interruption signal that in turn switches on the main radio transceiver to process the incoming data packet, and then, to go back to the sleep mode, is insignificant.

Increasing the number of SNs in the M2WSNs (N) has little effect, as shown in Figure 4.6b, that implies that the total size of the M2WSNs does not affect the effectiveness of a particular protocol scheme, if the other parameters are constant. However, the performance of the wake-up protocol based on WuRx decreases as N increases, due to the energy consumed by



(a)

(b)

(c)

(d)

Fig. 4.6.: The ratio of the total energy consumption between both schemes under different use cases.

non-destination SNs that remain awake when an event occurred and until they realize that they are not the destination node. In this situation, it might be beneficial to use a WuRx with addressing support and a false wake-ups reduction strategy. For higher number of hops (h) (Figure 4.6c), the protocol based on WuRx performs better than based on LDC and becomes beneficial, thanks to the WuRx operations that allow SNs within the routing path to remain less time waiting for the next hop to wake up when compared to LDC schemes. Finally, for short event periods ($mtbe$), WuRx scheme is better but somewhat affected by long event periods, due to the always-on operation assumed for the WuRx (Figure 4.6d).

A particular behavior in the ratio of WuRx/LDC can be analyzed when $h = 200$ and $N = 200$ (Figure 4.6c and Figure 4.6b), where resulted a ratio of 0.72 and 0.99, respectively. In both cases, $N = 200$ but in Figure 4.6b was configured to $h = 5$ (fixed) for the sensitivity study. When reviewing equations (4.8) and (4.13) that depend on equations (4.2) and (4.9), respectively, it can be evidenced that (4.2) and (4.9) are the ones that contribute most into the energy difference between the WuRx and LDC models, since they depend on $P_{\triangle DLPM}$ ($87.1mW$) and P_{node} ($108.9mW$), respectively. Hence, an LDC scheme adds an additional energy consumption of $21.8mW$ per hop to the total energy consumption of the network compared to the WuRx. Therefore, the LDC contribution is greater than WuRx when the ratio is performed, resulting in a lower ratio value in Figure 4.6c compared to the case presented in Figure 4.6b where $h = 5$.

4.4 Remarks

In summary, this chapter provides an energy consumption estimation model that considers the behavior and performance of wake-up protocols based on WuRx and the traditional LDC schemes employed in M2WSNs and multi-hop communications.

M2WSNs based on an always-on WuRx scheme save significantly more energy when compared to "classic" LDC schemes when using many hops on the routing path and for short event periods. Besides, the WuRx with addressing does not seem to save more energy when compared to WuRx without addressing, under a traffic load with short event rates. However, for circumstances of long event rates, the WuRx scheme might lose against an LDC approach regarding idle-listening. Duty-cycling the WuRx could be considered as a possible solution to this issue [17], but it might imply additional latency on delivery the packet towards the sink. An ultra-low power WuRx in the order of nanowatts could reduce even more the energy consumption of the whole M2WSNs during circumstances with long event rates. However, in the moment of this writing, nanowatt WuRx is in its prototype phase [104].

In the energy consumption model based on WuRx, we considered the worst case, i.e., where all SNs within a neighborhood into the M2WSNs are woken-up when a source node has an event packet to transmit towards the sink. Therefore, the energy consumed due to false wake-up is implicitly considered in the model, i.e., equation (4.4) for non-designated SNs.

We focused mainly on modeling the ON and OFF transitions of both approaches at the lower layers regarding energy consumption, providing a simple energy model that allows us to get essential insights about wake-up schemes for M2WSNs. Hence, for the sake of simplicity, the proposed model was designed considering some assumptions (see section 4.2.1) such as an error-free channel that does not imply additional data packet transmissions due to packet loss. A packet loss usually implies that the packet should be retransmitted more than once at the cost of a higher power consumption and latency [83]. The expectation of successful event-packet transmitted to the sink is usually modeled as a factor that follows an error distribution probability [95]. However, we should expect that the overall energy consumption increase, but in general, the behavior (ratio between WuRx/LDC) should remain the same.

Finally, we list again the basic assumptions considered in the model with their correspond analysis when each one is not considered as an assumption.

1. **The WuRx and the main radio share both the same channel. As a consequence, transmissions of the main radio are detected by the WuRx. Therefore, each sensor node needs, besides the main radio, only a WuRx, and not a wake-up transmitter**. If this assumption is not valid anymore, we should consider a transmitter to emit the WuS that might add some complexity on the hardware and additional energy consumption. Besides, for this case, the WuS usually consists of a simple non-data packet with the destination WuRx address. Hence, the receiver usually emits an acknowledgment for the WuS in order to receive the data packet that not only increases the power consumption but also the latency might be slightly affected [83]. For the energy model in question, it should be add a factor with relating the power consumption of the WuR transmitter and the time required to generate and to transmit the WuS. However, in the end, we should expect that the overall energy consumption increase, but in general, the behavior should remain the same.
2. **Both radios have the same communication range**. In our model, we considered that all nodes within the known path can only reach their parents. Hence, a multi-hop communication is considered for relaying the data packet towards the sink. However, there are scenarios where exists a mismatch between the WuRx and the main radio's range, because the WuRx are usually designed with lower sensitivity compare to the main radio, for energy savings purposes that imply that the destination is not reachable via the WuR but via the main radio. One solution is to increase the power of the WuR transmitter but at the cost of a higher energy consumption. Therefore, multi-hop communication is a feasible solution to overcome such situation [83] that was considered in the model.
3. **We do not care about the network topology and routing protocols. We assume that a sensor node knows the address of the next hop for delivering a data packet to the sink**. Topology maintenance and routing tasks mean additional messages exchanged between nodes within a neighborhood (e.g., keep alive messages). Extra packet transmissions represent additional energy consumption, but the ratio between both wake-up schemes should be maintained. Please, consider the example presented in the next point.
4. **Transmitted data message is very small. Therefore, we assume that it is enclosed in the wake-up packet**. In our model, we considered a packet size of 32bytes@100kbps. However, the IEEE 802.15.4 technology supports an MTU of 127bytes@250kbps. One expects that higher data messages mean more time to transmit at the cost additional power consumption. Therefore, the overall energy consumption in both models will increased when a data message is transmitted toward the sink, but the ratio between WuRx and LDC might remains similar. For instance, consider

the parameters configuration used for the sensitivity study (Table 4.4: $mtbe = 60s$, $T_{sleep} = 10s$) and $N = 200$, $h = 5$, $nbor = 50$. One data message (32bytes@100kbps) takes $T_{data} = 2.56ms$ to be transmitted over the air (Table 4.3). Under these conditions, the ratio calculated is 0.993. Now, assuming the data message has twice its size (2 packets@32bytes). Consequently, $T_{data} * 2$ is needed for transmitting it. Hence, the ratio is maintained, i.e., 0.989. For five times the size (5 packets@32bytes), $T_{data} * 5$ required, then, the ratio is 0.978. Therefore, the ratio between WuRx and LDC is slightly affected, remaining the same for the first significant figure, when the time for transmitting a data message increases.

5. **For the sake of simplicity, we assume that the power consumption (P_{node}) of the main radio is the same for the wake-up and idle periods, receiving and transmitting a packet, and switching between states**. Higher or lower power consumption values employed in the model might increase or reduce the overall energy consumption, but one expects that the ratio WuRx/LDC remains unchanged, without affecting the final conclusions.
6. **For energy estimation, we assume no packet loss during communication, i.e., an error-free channel**. A packet loss usually implies that the packet should be re-transmitted more than once at the cost of a higher power consumption and latency [83]. The expectation of successful event-packet transmitted to the sink is usually modeled as a factor that follows an error distribution probability [95]. However, we should expect that the overall energy consumption increase similarly in both models [83], maintaining the ratio between WuRx/LDC, without affecting the final conclusions.
7. **All SNs sleep in the deepest operating sleep mode, until they are woken up for their active period**. In case a different LPM is used, we expect that power consumption slightly increases. Besides, if the LMP0 is employed, all peripherals are still operational at the cost of an additional energy consumption (from active mode to LPM0, the energy draws from 300μA to 55μA in an MSP430 micro-controller at 1MHz@3.3V) compare to the LPM4 where the CPU and all timers are disabled, causing a current reduction up to 0.1μA [99, 100].

5

A MultiModal Switching Mechanism for Data-Gathering Schemes

The M2WSNs arise as a solution for the trade-off between energy savings and event reporting latency in those monitoring oriented applications where regular and emergency reporting are required simultaneously, such as forest fires and pollution monitoring. The multimodality in these M2WSNs allows sensor nodes to perform data reporting in two possible schemes, time-driven and event-driven, according to the circumstances, providing higher energy savings and better reporting results when compared to traditional schemes. In this chapter, we present the implementation and evaluation of M2-DaGNoS, an enhanced MultiModal Switching mechanism for Data Gathering and Node Scheduling, suitable for monitoring applications that simultaneously manage normal and emergency circumstances. This technique is implemented in ContikiOS, an open-source operating system for the Internet of Things (IoT), and evaluated using the Cooja emulator through an extensive simulation study. Besides, the operation of M2-DaGNoS is validated and evaluated in a laboratory-based environment under different experiments. All performance evaluations have been performed under a single-radio architecture.

5.1 Overview

As mentioned in Chapter 1, in M2WSNs, there is a trade-off to be made between reporting accuracy and energy-saving regarding the data reporting schemes, i.e., CMnt and EDR. We require to implement a multimodal switching mechanism between CMnt and EDR modes that combines the advantages of both schemes. In this chapter, we present the design and implementation of M2-DaGNoS, a multimodal switching mechanism suitable for monitoring applications that simultaneously manages normal circumstances and emergencies in M2WSNs. Besides, the M2-DaGNoS mechanism is implemented under a single-radio architecture. In Chapter 6, M2-DaGNoS is evaluated under a dual-radio architecture based on the WuRx paradigm.

In Table 5.1, a qualitative comparison is made between relevant M2SMs proposed in the literature and our approach. This comparison is made based on some design features such as data-reporting frequency (i.e., if upon the occurrence of an event, the frequency remains fixed or it is changed), data-reporting format (i.e., if the payload included into a packet is reported without modifications (raw data) or as the average of several measurements of

Tab. 5.1.: Comparison of relevant M2S mechanisms.

Mechanism	Data-reporting frequency/format	Multi-hop communication	Route & Neighbor discovery	Node queue building	LDC	WuRx	PED	Network time synchronization
HDR [6]	Fixed, average	✓	×	×	×	×	✓	×
HNS [25]	Fixed, raw	✓	×	1st WN	×	×	×	×
eHNS	Fixed, raw	✓	✓	1st WN	✓	×	×	×
M2-DaGNoS	**Variable, average**	✓	✓	Every SN	✓	✓	✓	*Implicit*

the physical variable), multi-hop communication (i.e., if packets are forwarded toward the sink via a multi-hop path). Besides, the comparison considers those techniques supported by the mechanisms such as routing and neighbor discovery, node-queue building and parameter-based event detection (PED), wake-up protocols (LDC, WuRx), and network time synchronization.

The studies in Table 5.1 have been selected, from those previously revised in the literature review conducted in section 2.3, since the authors provided sufficient details of their design, facilitating the processes of coding and implementing their proposals, from scratch, in ContikiOS. In addition, the studies followed a design approach from the application layer perspective as M2-DaGNoS mechanism, compared to the work proposed in [13], that focused on the physical layer using digital signal processing tools. Finally, the M2-DaGNoS mechanism, along with the proposal in [13], has considered a WuRx in its design process (see Chapter 6).

The M2-DaGNoS mechanism, in comparison to the other approaches in Table 5.1, employs a variable frequency for data reporting according to the circumstances, where the data reported are the average of several measurements of the physical variable during a time window. Besides, an implicit network time synchronization is included for data sensing and communications. In addition, a low-duty-cycling scheme and power-aware network mechanisms are combined with the M2-DaGNoS for further energy savings during the whole life of the nodes. Finally, it includes an enhanced version of the cyclical node sleep scheduling mechanism proposed in the HNS mechanism [25], combined with the PED algorithm provided in HDR [6], to manage the goals of both CMnt and EDR schemes. The eHNS mechanism is an enhanced version of the *HNS* technique proposed in [25] – details of the implementation process is described in the Appendix A.

In the following sections, we describe the design approach of M2-DaGNoS at the application layer. Then, we present and propose the power-aware mechanisms at the lower layers. Finally, the performance evaluation and results through emulations and laboratory-based experiments are provided. Hence, the main contributions of this chapter are summarized as follows: (1) An enhanced multimodal switching mechanism for data gathering and node scheduling in M2WSNs monitoring applications is proposed and evaluated. (2) A comparison is made between M2-DaGNoS and state-of-the-art switching mechanisms through

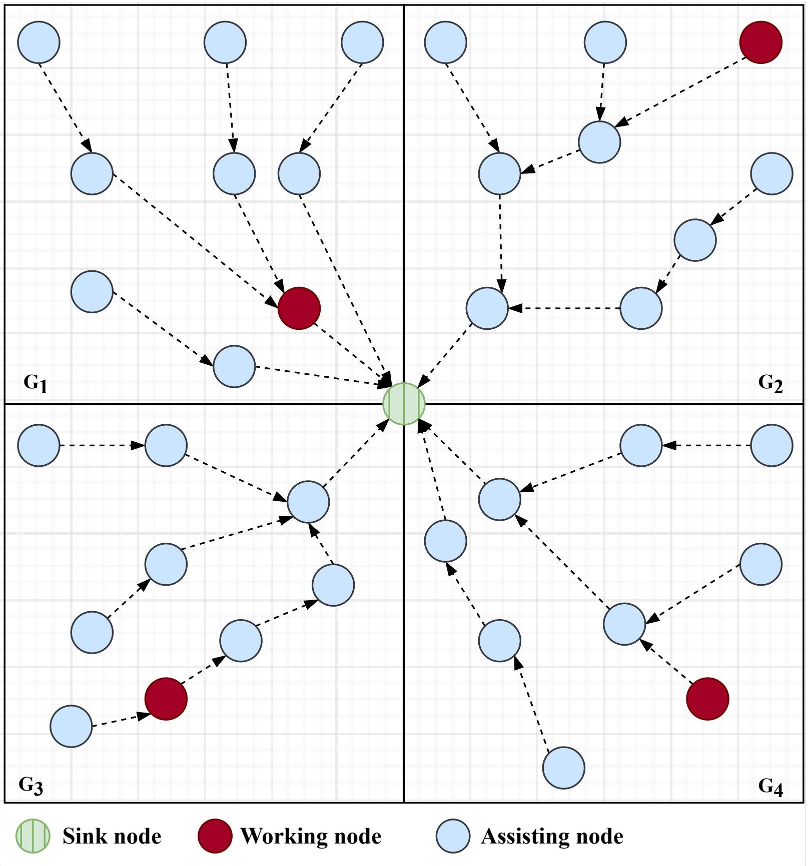


Fig. 5.1.: Example of M2WSNs divided into four grids that are adjacent to the sink node. Each grid includes one working node and several redundant or assisting nodes.

an extensive emulation study under a single-radio architecture. (3) A validation of the M2-DaGNoS operation via hardware experiments is performed.

5.2 M2-DaGNoS Design

M2-DaGNoS is designed based on the assumption that a supervising area is divided into grids (i.e., clusters), where a single node will always be working in each grid during normal circumstances, as shown in Figure 5.1. Having one working node per grid increases the possibilities to the M2WSNs to monitor all grids and be aware of an event occurrence under emergencies [25]. During an emergency, the working node is assisted by the other nodes in the grid (assisting nodes) that report more information about the occurrence of an event for tracking purposes. The working node performs duties such as sensing, data reporting, event detection over a scheduled period. The role of the working node is cyclically rotated between nodes of a grid for energy-savings, coverage guarantee, during a scheduled cycle.

In Figure 5.1, within each clusters in the M2WSNs, smaller grids are representing the use of a higher spatial resolution mode to provide better monitoring results during normal circumstances and to spatially track an event during emergencies. It is assumed that SNs can set on a higher spatial resolution mode during their working and assisting operations and a lower spatial resolution during the sleeping mode for most extended sensing ranges.

Additional underlying assumptions are made and read as follow:

1. The M2WSN is divided into several grids by a previous clustering procedure. As a result, some grids are adjacent to the sink node (Figure 5.1)[1] Non-adjacent grids are not considered in this work[2].
2. Each adjacent grid constructs and maintains a minimum spanning tree [105], having a common sink node as the root of the tree (Figure 5.1).
3. All nodes within a grid can communicate with each other[3]. Therefore, a node queue is constructed, and M2-DaGNoS messages (i.e., working and assistant messages) are exchanged between nodes.
4. All grids work under the same RF channel and CSMA scheme, but each grid maintains different logical connection channels to transmit unicast and broadcast messages. Therefore, inter-grid communication collisions during the M2-DaGNoS operation are mitigated and extra packet receiving procedures in the adjacent grid are avoided.
5. Within a grid, there are also some redundant nodes in sleep mode. The sleep mode is controlled by a radio duty-cycling technique at the link layer. In this mode, the nodes are not in their lowest power mode (e.g., LPM4), the microcontroller remains active and the radio is managed by a low-duty-cycling technique (e.g., ContikiMAC).

M2-DaGNoS helps to reduce the power consumption of M2WSNs by implementing two methods: node scheduling and data gathering. The former is an improved version of the cyclical node sleep scheduling method proposed in [25]. The latter combines the capabilities of the event-driven node wake-up method developed in [25] and the PED algorithm introduced in [6].

---

[1] To be an adjacent grid, at least one node must have direct communication to the sink node (one hop). The rest of the nodes send packets towards the sink through the collect tree built during the neighbor discovery process.

[2] Non-adjacent grids work similar to the adjacent grids, but they transmit data packets towards the sink through the adjacent grids using a power-aware multi-hop routing mechanism [105]. Inter-grid communications are done via the same RF channel, but under a different logical connection channel for message exchanging. Later in section 5.5, more details are provided.

[3] A single-hop or multi-hops communication might be used, depending on the current routing metric used to select the parent in the tree [89].

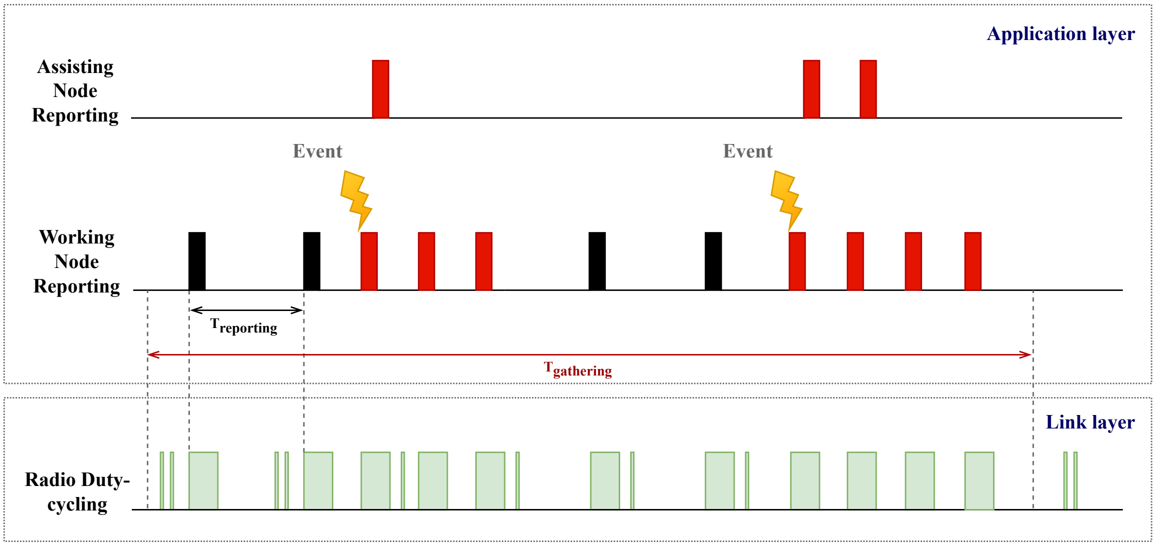


Fig. 5.2.: Data reporting and radio duty-cycling during a scheduling cycle. Red and black blocks, at the application layer, indicate packets reported to the sink by working and assisting nodes during an normal circumstance and an emergency, respectively. The green blocks and their lengths represent the active period of the main radio, period managed by a radio duty-cycling technique at the link layer.

The M2-DaGNoS has been designed to work above the network layer at the application layer. From the radio perspective, the M2-DaGNoS approach helps to minimize the power consumption by managing a data reporting duty-cycling at the application layer, using the methods mentioned before, combined with a radio duty-cycling at the link layer as shown in Figure 5.2. At the application layer, a fixed data reporting duty-cycling has been considered. However, multiple data reporting duty-cycling can be set dynamically. For instance, based on the event behavior under supervision, it might imply the need of additional algorithms (e.g., complex machine learning algorithms).

Figure 5.2 shows the duty-cycling operation performed by nodes at the application layer (using the M2-DaGNoS mechanism) and at the link layer (implementing a radio duty-cycling technique [50]). Every working node propagates packets during a scheduled period ($T_{gathering}$) using a defined data-reporting frequency ($T_{reporting}$). The data-reporting frequency is changed during an emergency to report more data associated with the occurrence of an event (e.g., $T_{reporting}/2$). The black blocks indicate those packets reported to the sink by a working node during a normal circumstance. The red blocks represent those packets generated during an emergency by working and assisting nodes. The radio duty-cycling technique at the link layer manages the active period of working and assisting nodes' radio to check any channel activity (shorter green blocks), receive or transmit a data packet (longer green blocks).

Algorithm 2 Node Scheduling

```
Input: Node queue of size $N_j$, where $j$ is the $id$ of the $j$th grid (i.e., $G_j$); $nbor$: neighbors discovered; $s_i$: $i$th sensor node id.
Output: Node queue Q, working node turn $\gamma \geq 1$.
procedure NODE SCHEDULING($N_j$, $nbor$, $s_i$)                      ▷ For each $s_i$ in $G_j$
    Neighbor discovery initialization
    Read $nbor$                  ▷ Do until all neighbors are discovered (i.e., $N_j - 1$)
    if $nbor = N_j - 1$ then
        Get Neighbor table NT
        Sort elements of NT in ascending order
        Store elements sorted in a new table Q
        for all element $q_k$ in Q do                             ▷ Go through Q, $k++$
            if $q_0 = s_i$ then                                   ▷ $q_0$ first element of Q
                Select $s_i$ as the first WN with $\gamma = 1$
                Save next WN id $q_1$
                Set $t_{work}$
            else
                Select $s_i$ as AN with $\gamma = k$
                Save next WN id $q_{k+1}$             ▷ If $k + 1 = Nj \rightarrow$ Next WN id $q_0$
                Set $t_{sleep}$                         ▷ $t_{sleep} = (k - 1) * t_{work}$ [25]
```

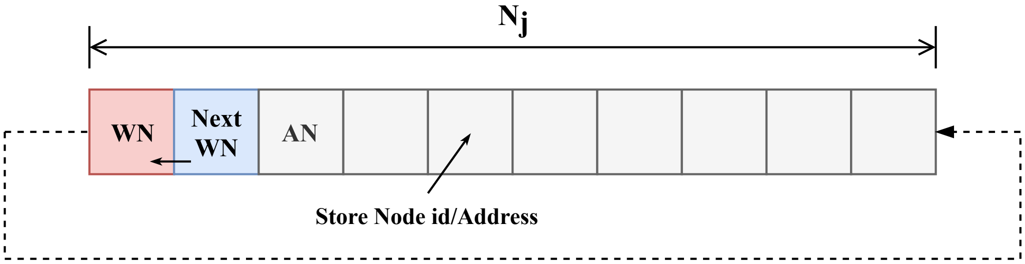


Fig. 5.3.: A node queue **Q** of size $N_j = 10$ of the grid G_j, where the first position is for the first working node (WN), the second for the next WN, and the remaining positions for the assisting nodes (AN) within G_j. Each node in G_j stores the full node queue during its neighboring discovery procedure.

5.2.1 Node Scheduling Mechanism

One grid is composed of several sensor nodes and, inside each grid, nodes work according to a scheduling mechanism (see Algorithm 2). This mechanism keeps one node in a working state and the rest of nodes in a sleep state until it is their turn to be a working node (WN) or become an assistant node (AN) to the current WN during the occurrence of an event within the grid. During the initialization of the network, all nodes in the grid are in the same communication range. Therefore, message exchanges can be made between them within one hop.

Each WN remains active to guarantee coverage and connectivity during a scheduling cycle, t_{work}. The radio connectivity is given by the radio duty-cycling approach at the link layer (see Figure 5.2). The working role is rotated among all SNs that belong to the same grid to avoid the rapid depletion of the battery and not having any areas without being monitored for a long time. This role is assigned to one node at a time, taking turns based on its position within the "node queue" **Q** that is constructed during the settlement of the network (see Figure 5.3). Additionally, the information on the node queue is used by the current WN to know which node should wake up next to perform the working role. Therefore, each node within a grid is scheduled to wake up one after the other and perform data gathering duties (e.g., sensing and reporting) during a t_{work} period[4].

The node queue **Q** is built using the information collected by each node during its neighbor discovery procedure performed on the settlement of the network. Initially, SNs do not know who are their neighbors because they are usually randomly deployed in the area of interest. Thus, each node periodically collects information about their neighbors by exchanging discovery messages with each other. Then, each node builds and maintains a neighbor table for routing and packet forwarding duties [105]. M2-DaGNoS uses the information on the neighbor table (e.g., node id or address) to build the node queue. Once the neighbor table is completed (after the initialization of the network), the node scheduling algorithm requests the information available on this table to construct the node queue that is a new table but its elements are sorted by applying a simple criterion: sort the elements by neighbor identification from the lowest to the highest id. Afterwards, each node gets its working turn, $\gamma \geq 1$, based on its position on the node queue **Q** and transits between working and sleeping states according to a scheduling cycle (within a grid G_j) that satisfies the following condition [25]:

$$T_{scheduling} = k \times t_{work} \tag{5.1}$$

where the constant k is defined as the number of neighbor within a j-th grid, G_j, i.e., $k = m \cdot N_j$, $m \geq 1$ an integer. t_{work} is the working time of a node defined in terms of the time spent during a data-gathering cycle, i.e., $t_{work} = m \cdot T_{gathering}$.

As mentioned before, the node queue is only built during the network initialization, using the information provided by the neighbor table that is populated and maintained during the whole operation of the M2WSNs using the neighbor discovery mechanism available on the collect protocol implemented at the network layer [90]. It is assumed that the nodes within the grid will not fail throughout the operation of the M2WSNs. However, in case of failure, the node queue can be updated using the information available on the neighbor table that is

---

[4] In case the WN runs out of energy during a scheduling cycle, case which is not supported by the current version of the scheduling algorithm, a future improvement of the scheduling algorithm might be the integration of the node's residual energy (e.g., the voltage) as input to decide when it is appropriate to communicate the withdrawal decision to the next node in Q or the neighborhood, to avoid or mitigate the problem of not monitoring the area due to lack of energy.

maintained by the collect protocol. In the current version of M2-DaGNoS, the node queue remains the same for the whole operation of the M2WSNs. However, a future improvement might be to implement a procedure that keeps the information within the node queue up to date using a cross-layer approach between the application and network layers.

The neighbor discovery mechanism is based on repeated broadcasts as described in [90] that periodicity varies during the initialization of the network but after reaching a maximum interval, the periodicity remains fixed. This condition might affect the overall performance of the network. However, the effect of periodic broadcasts might be mitigated by selectively setting the maximum interval according to the link condition or using partial and bounded updates, i.e., send beacons only when there is a change on the topology and only to those nodes that are affected and need that information (e.g., a parent node that has several children over the tree-like topology).

To sum up, M2-DaGNoS takes advantage of the neighbor discovery procedure to build the node queue during the initialization of the M2WSNs, presenting better management results when compared to the method in [25], where the information on the node queue is only stored on the first WN and then transferred to the next WN when it wakes up to work.

5.2.2 Data Gathering Mechanism

M2-DaGNoS provides a data gathering mechanism (see Algorithm 3) that allows SNs to combine the ability to react immediately upon the occurrence of a known event (based on a threshold criteria (e.g., $T \geq 100°\text{C}$)) and to track it using the capabilities of an event-driven data-gathering scheme (i.e., EDR). Under normal circumstances, SNs report sensed data using continuous monitoring capabilities (i.e., CMnt). In this way, M2-DaGNoS improves the reporting accuracy, provides up-to-date data to the sink about the area of interest and corresponding events, and extends the lifetime of the M2WSNs.

The switching between EDR and CMnt data-reporting modes and vice versa is performed dynamically based on the results given by the PED algorithm [6]. The PED algorithm computes the current average of the sensed data, S_{cur}, collected by a node during a recent time window[5]. Then, it compares this value with a threshold value, Th and the previous average sensed data, S_{prev}. Afterward, it increases two counter-variables, Γ and Λ, whether the comparisons are satisfied. Otherwise, the counter-variables are reset to avoid the risk of transient response. Finally, the two counter-variables are compared with two-tuple of

---

[5] An average sensing-based method has been employed for the data reporting that might be affected by some abnormal peaks in the sensing values, that can represent some relevant issues in the monitoring variable presented in some practical monitoring applications. However, to handle this issue, a differential-based method can be considered instead, such as the one used in audio coding, e.g., delta compression or adaptive differential pulse code modulation.

Algorithm 3 Data Gathering

```
Input: Constant $tuple\{\Gamma, \Lambda\}$; $Th$, threshold value; $\Delta = \{\delta_{min}, \delta_{max}\}$, number of packets
  (granularity) to report; $S_{cur}$, current average sensed data.
Output: Reporting mode $M$, Reporting frequency $\delta$.
procedure DATA GATHERING($Th$, $tuple\{\Gamma, \Lambda\}$, $\Delta$)
    Sensing data procedure initialization
    PED algorithm initialization                              ▷ $PED(S_{cur}, Th, tuple\{\Gamma, \Lambda\})$
    while $t_{work}$ not expired do                           ▷ Only apply to working nodes
        Check PED algorithm result                            ▷ After checking the timer associated
        if Equation (5.2) is satisfied then
            Set $M$ to EDR and $\delta = \delta_{min}$                            ▷ Just once
            Generate and send event-data reporting packets to the sink
            Broadcast an assisting message to all nodes within $G_j$          ▷ Just once
        else
            Set $M$ to CMnt and $\delta = \delta_{max}$                           ▷ Just once
            Generate and send data reporting packets to the sink
        if $t_{work}$ expired then
            Transmit a working message to the next WN
            if $S_{cur} \geq Th$ then
                Stay alert as an assisting node within $G_j$
            else
                Go back to sleep
    if Assisting status then                                  ▷ Only apply to assisting nodes
        Set $\delta = \delta_{min}$                                               ▷ Just once
        Generate and send data reporting packets to the sink
        if $S_{cur} < Th$ then
            Set $t_{sigma}$                                                   ▷ Just once
            if $t_{sigma}$ expired then
                Go back to sleep
```

constants parameters: $(\Gamma_{start}, \Lambda_{start})$ and $(\Gamma_{stop}, \Lambda_{stop})$ that define two switching conditions, (5.2) and (5.3), depending on the current data-reporting mode.

If condition (5.2) is satisfied, i.e., S_{cur} is above Th and increases (i.e., $S_{cur} \geq S_{prev}$) during Γ_{start} or Λ_{start} consecutive time periods, then the node switches to EDR mode. Likewise, if condition (5.3) is satisfied, i.e., S_{cur} is below Th and decreases (i.e., $S_{cur} < S_{prev}$) during Γ_{stop} or Λ_{stop} consecutive time periods, then the node switches to the CMnt mode. Otherwise, the node remains in its current mode.

$$\text{Current mode CMnt: } \Gamma \geq \Gamma_{start} \text{ or } \Lambda \geq \Lambda_{start} \tag{5.2}$$

$$\text{Current mode EDR: } \Gamma \geq \Gamma_{stop} \text{ or } \Lambda \geq \Lambda_{stop} \tag{5.3}$$

The data gathering mechanism works as follows. Under normal circumstances, a working node starts in a continuous monitoring mode, i.e., it reports sensed data to the sink every $T_{reporting}$ (a predefined reporting time). The sensed data reported is computed as the average of several sensed values collected over a recent time window. Upon the occurrence of a known event (under an emergency), the WN switches to the event sampling mode, based on PED results, generates an event reporting packet and quickly transfers it towards the sink. Then, the WN informs subsequent changes to the sink with a higher reporting frequency (e.g., $\delta_{min} = T_{reporting}/2$) and notifies its neighborhood (within the same grid) to continuously report sensed data as well (for event tracking purposes). The latter is done by implementing the event-driven node wake-up mechanism proposed in [25].

The event-driven node wake-up method is used to better characterize and track the event of interest from the sink side. This mechanism is performed as follows: (i) A WN perceives a known event in the area of interest (based on PED results); (ii) The WN generates and sends an event reporting packet (i.e., an alarm) to the sink; (iii) For better event trend tracking, the WN broadcasts an assisting message to all redundant nodes within the grid; (iv) The redundant SNs (assisting nodes) transit from a sleeping state to an assisting state after woken up by the WN, and begin to continuously disseminate sensed data to the sink every certain time (see the latter process in Figure 5.2); (v) These assisting nodes go back to sleep if their sensed data are below a threshold (e.g., $T \geq 100°$C) after a predefined interval of time t_{sigma}. The last step is done to save energy and to reduce the network overhead [25].

5.2.3 Principle of Operation

The M2-DaGNoS mechanism is modeled using a state machine approach as shown in Figure 5.4. From an application perspective, M2-DaGNoS goes through different states according to the scheduling cycle and the node queue information. All nodes start at the *Node Scheduling* state, where they perform a neighbor discovery process, build their corresponding node queue, and wait for t_{start} time interval to finish the settlement of the network. Afterward, the node with the lower identification is claimed as the first working node in the grid and transits to the *Working* state. The remaining SNs within the grid, the redundant nodes, go to the *Sleeping* state and stay there until their turn to be a WN or receive an assistant message from the current WN.

The redundant nodes switch from the *Sleeping* or *Assisting* states to the *Working* state when they receive a working message from the current WN. Additionally, the redundant nodes can also switch from the *Sleeping* to the *Assisting* state after receiving an assistant message. They return to the *Sleeping* state if their current sensed data is below a threshold value (i.e., $S_{cur} < Th$) during a t_{sigma} period (for energy savings purposes).

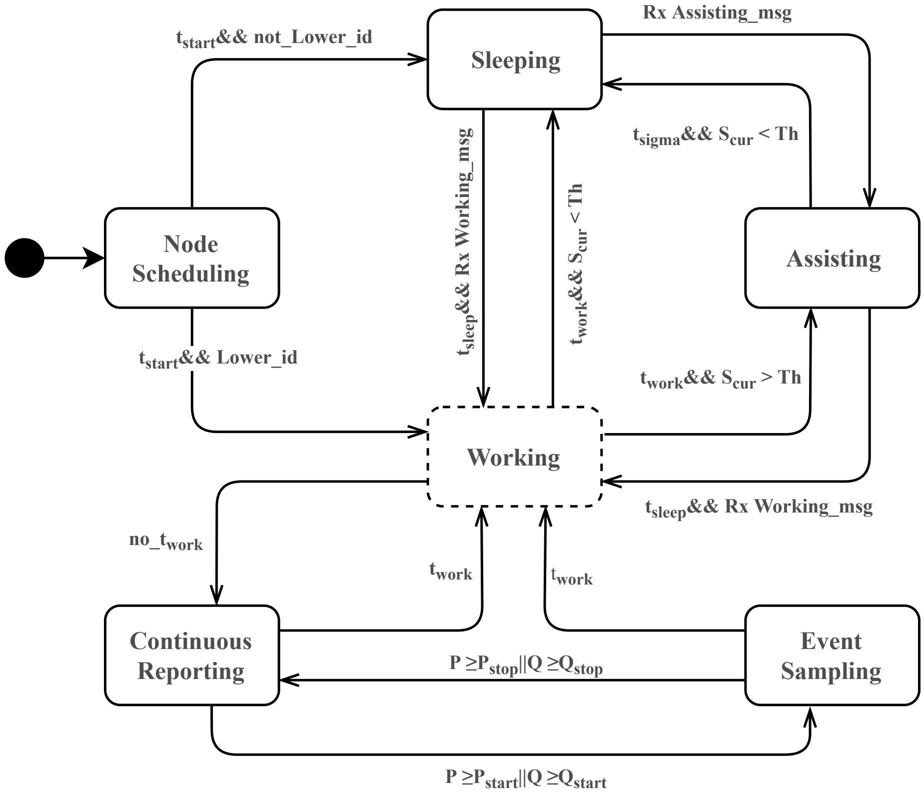


Fig. 5.4.: M2-DaGNoS state machine model.

The WN can transit to different states depending on its current state and the actions that have occurred. Initially, all WNs switch from the *Working* state to the *Continuous Reporting* to perform CMnt duties during a working period, t_{work}. Afterward, under normal circumstances, it passes through the *Working* state to the *Sleeping* state.

Under emergencies and based on the PED condition (5.2), the WN switches to the *Event Sampling* state to perform event-driven reporting duties. After the known event fades, the WN returns to the *Continuous Reporting* if the PED condition (5.3) is satisfied. If t_{work} expired, it goes through the *Working* state to the *Sleeping* state or to the *Assisting* state if the event has not faded.

5.2.4 Power-Aware Techniques and Communication Protocol Stack

The communication protocol stack implemented for M2-DaGNoS mechanism under ContikiOS is shown in Figure 5.5. ContikiOS [106] is an open-source operating system wrote in C language and supported by a worldwide developer community.

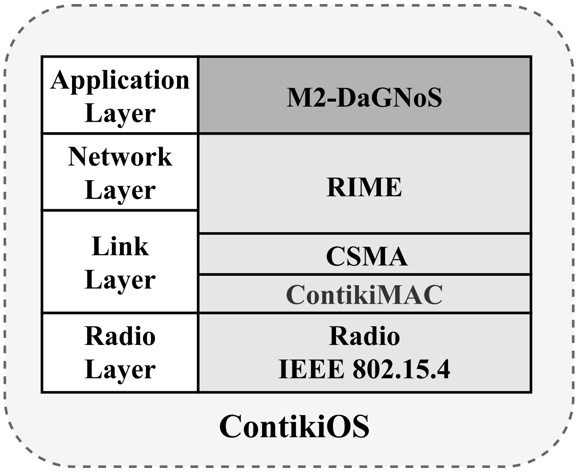


Fig. 5.5.: Communication protocol stack employed in M2-DaGNoS approach.

The application layer is in charge of performing techniques such as multimodal switching, node scheduling, and data gathering, using the information provided by its adjacent layer. The network layer is responsible for performing topology management duties, i.e., neighbors discovery, path selection and packet forwarding, network time synchronization. Here, we have implemented the well-defined network protocol stack available in ContikiOS, RIME [89], because it offers us robust ready-to-use modules such as broadcast, unicast, multi-hop, collect tree protocol, route and neighbors discovery modules–modules widely chosen in real WSNs applications. To synchronize the clock of all SNs in M2WSNs, we have implemented the implicit and periodic sender/receiver-based pairwise synchronization technique proposed in [107] and supported in ContikiOS. The term *implicit* refers to the fact that this technique does not require specific time synchronization messages.

The link layer performs RDC and MAC duties. We have employed an asynchronous low-duty-cycling technique, ContikiMAC [50], available in ContikiOS as a RDC mechanism. We have chose ContikiMAC because of its lower latency and duty-cycle (i.e., lower power consumption), and its higher packet delivery ratio (PDR) (i.e., less retransmissions) [51]. For MAC duties, we implemented a CSMA/CA strategy available in ContikiOS. Finally, at the physical layer, a lower-power RF chip driver is employed (e.g., cc2420[6]) for data packet transmission and receiving procedures.

---

[6] The cc2420 is the RF chip used in the SkyMote/Telos B mote available in Cooja emulator. The SkyMote is widely employed for evaluating the performance of WSNs studies [83, 82, 107], giving its extensive support in ContikiOS and Cooja tools for power consumption trace such as Powertrace. Hence, the emulation study of M2-DaGNoS performance has been done using the cc2420 instead of cc1101 used in section 4.3. The current consumption of cc2420 is 17.4mA (0dbm) for data transmission and 18.8mA for data reception at 2.4GHz with a 3.0V power supply compared to cc1101, 16.8mA (0dbm) TX and 16.9mA RX @868MHz, 3.0V–technical information provided by the manufactures. Hence, under an LDC scheme, it is expected to introduce an increment in the total power consumption of SNs when cc2420 is used instead of cc1101, giving the contribution provided by the parameter P_{node} in equations (4.10) and (4.11), that increases from $111.21mW$ to $119.46mW$.

5.3 Emulation Study

In this section we present the setup of the emulation experiments and the corresponding results and analysis.

5.3.1 Methodology and Configuration Parameters

The emulation studies are performed in Cooja emulator. The main goal of these studies is to evaluate the functionalities and robustness of our M2-DaGNoS mechanism. It worth to mention that the code tested in Cooja is the same code uploaded to real motes. As a consequence, we can reduce the development time drastically when deploying real WSNs applications [2].

We follow the automated procedure proposed in Figure 5.6 that allows us to automatically run different experiments defined based on the radio medium, the density of nodes within M2WSNs, and the event occurrence rate. In Figure 5.7, we provide a general view of the main component considered for the experiments: radio medium, topology, and performance metrics. The M2WSNs can include few or scarce nodes (e.g., 10 nodes) to a more dense density of nodes per area (e.g., 20 to 40 nodes). The occurrence of an event can vary in time and space within the M2WSNs based on the temperature profile provided in Figure 5.8, and the location of the current working node, respectively. Later in Table 5.2, more information is provided.

We configure a typical M2WSN with a scarce and a dense number of nodes uniformly distributed within different $30m \times 30m$ rectangular grids (Figure 5.1). We assume one sink node with a permanent power source for the whole M2WSN. We inject in every working and assisting node a synthetic temperature profile shows in Figure 5.8 that mimics a forest fire phenomenon over time. This profile is followed in a sequential way over time. The switching moments between CMnt and EDR, according to the conditions (5.2) and (5.3), are shown in Figure 5.8. All SNs are emulated SkyMotes. Table 5.2 shows the most important emulation parameters used in all experiments performed during the study. These parameters were defined based on [6] and [25]. δ_{max} and δ_{min} are expressed as periods in seconds.

For the evaluation of M2-DaGNoS under dense M2WSNs, the M2WSN has been divided into four grids –adjacent to the sink– with a total of 40 SNs (see Figures 5.9a and 5.9b). A collect tree-based topology is built within each grid using the collect tree protocol available in RIME . The sink node is strategically located at the center of the area of interest where it is adjacent to all grids. Every grid has only one working node at a time and several nodes that work as assisting nodes during an emergency and remain in sleep mode during a normal circumstance (see Figure 5.1).

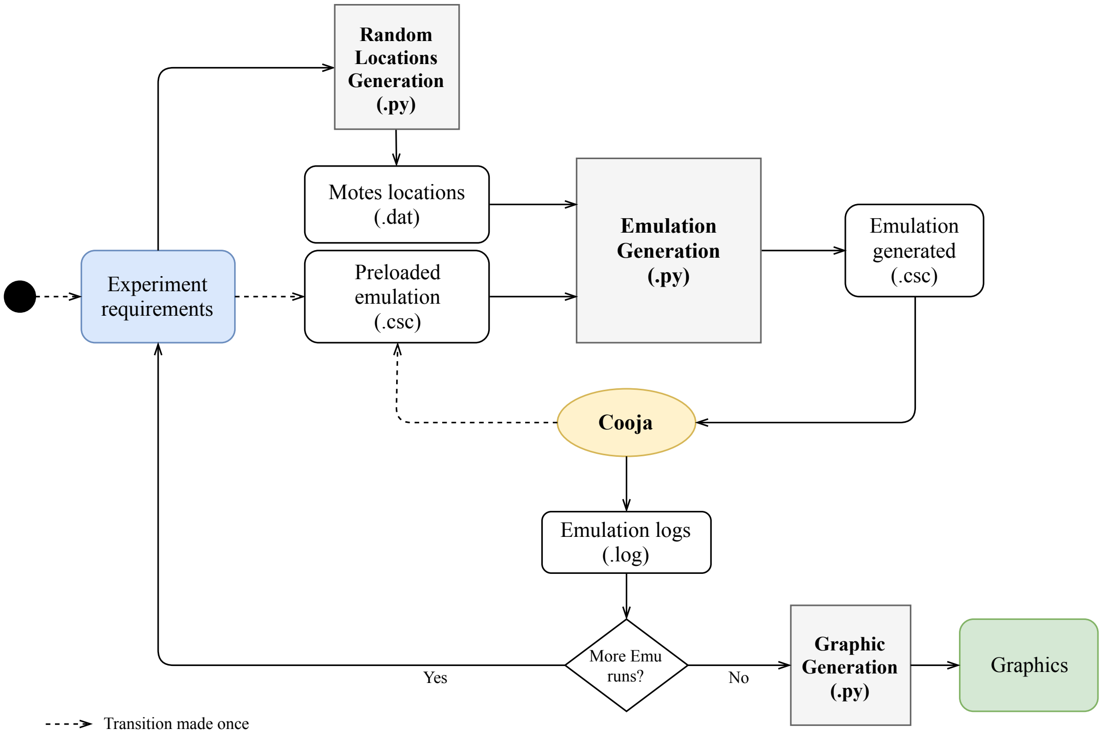


Fig. 5.6.: Automated emulation runs methodology.

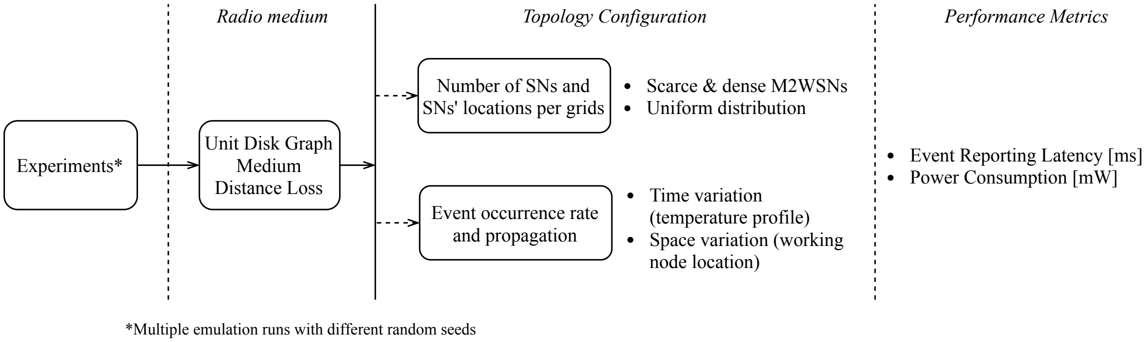


Fig. 5.7.: Experiments based on the radio medium, configuration parameters and the performance metrics for the emulation study.

For comparison and validation of the performance of the M2-DaGNoS technique, we have implemented three schemes that main features were described in Table 5.1:

1. *HNS*[25] is a hybrid node scheduling mechanism for M2WSNs implemented under an always-on scheme. We have implemented this mechanism from scratch in ContikiOS, given that the source code was not released by the authors.
2. *eHNS* is an enhanced version of *HNS* that utilizes always-on and duty-cycling schemes. Its implementation is described in the Appendix A.

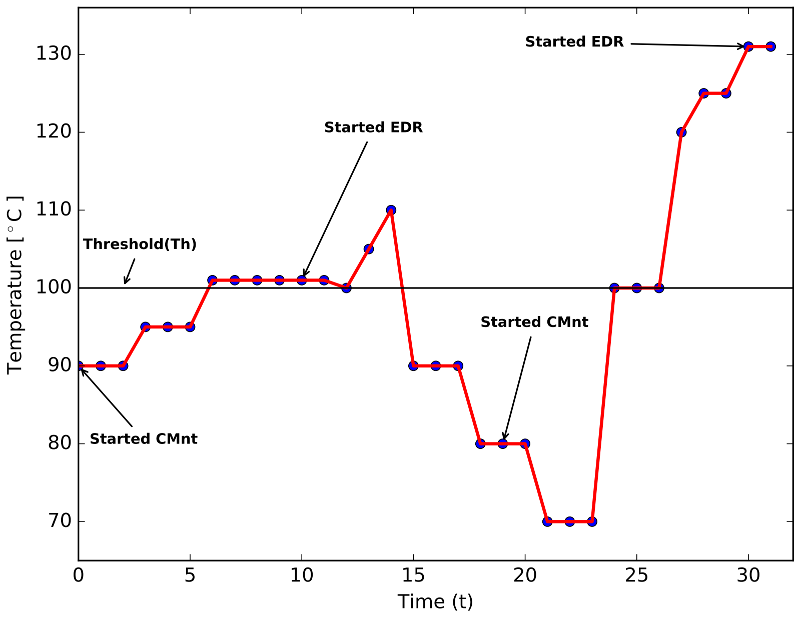


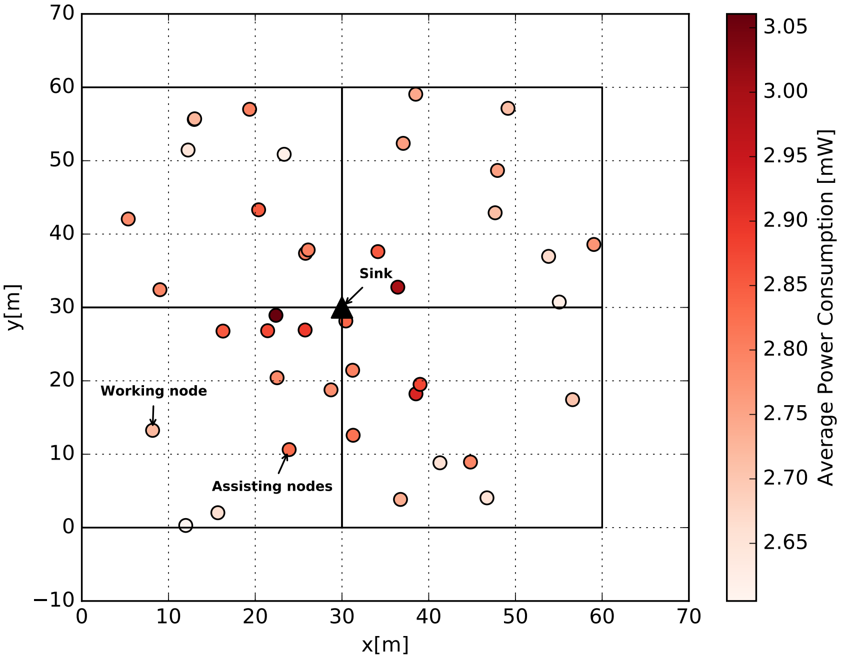
Fig. 5.8.: Synthetic temperature samples over time with switching moments between CMnt and EDR modes. This profile is followed by each node in a sequential way over time.

Tab. 5.2.: Configuration parameters for the emulation study (based on [6] and [25])

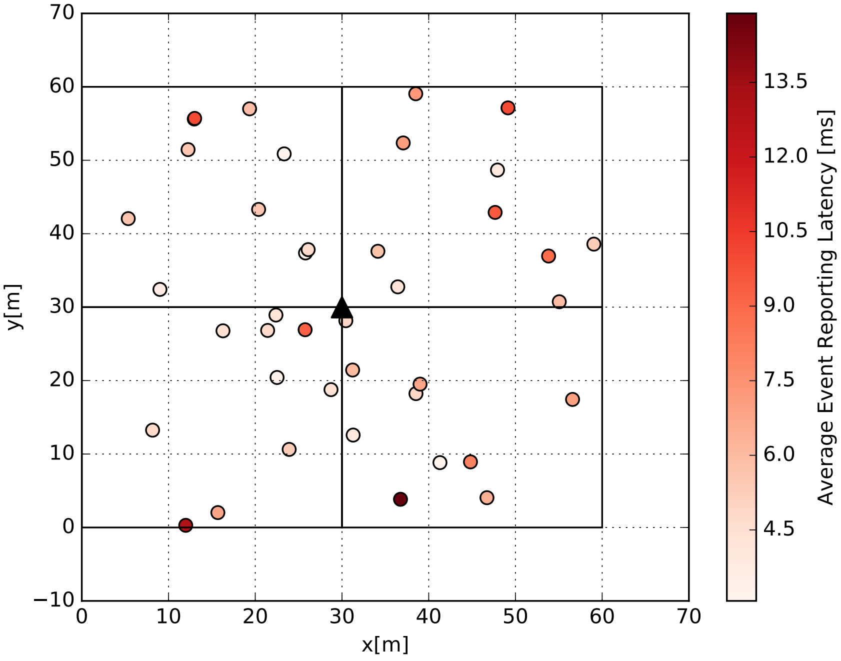
General Parameters	Value	General Parameters	Value
Grid number	1 to 4	Simulation time	35 min
Grid size	30x30m^2	Γ_{start}	3
Number of SNs in the M2WSN	10, 20, 40	Λ_{start}	5
Transmission range	50m	Γ_{stop}	3
Radio medium	UDGM: Distance Lost	Λ_{stop}	5
Event threshold (Th)	100°C	PED execution	5 secs
$\delta_{max} = T_{reporting}$	10 secs	t_{work}	50 secs
δ_{min}	5 secs	t_{sigma}	5 secs
Sampling rate	1 secs	t_{start}	120 secs
Payload (Data)	37 bytes	Payload (Ack)	5 bytes

3. *M2-DaGNoS* is the enhanced multimodal switching mechanism for data gathering and node scheduling proposed in this chapter.

The lower layers implemented in *eHNS* are the same as the proposed in the communication stack of *M2-DaGNoS* (see Figure 5.5). The HNS mechanism shares the same network layer as eHNS and M2-DaGNoS, but it differs at the link layer, where null MAC and null RDC are implemented, i.e., no MAC and no radio duty-cycling functions are implemented, as proposed in [25].



(a) Power consumption.



(b) Event reporting latency.

Fig. 5.9.: Example of a M2WSN divided into 4 grids with 10 SNs per grid and the sink located at the center of the area of interest. (a) and (b) are the network performance computed using the M2-DaGNoS mechanism. SNs closer to the sink have higher power consumption, but usually lower event reporting latency.

Finally, we define two quantitative metrics to analyze and compare the performance of M2-DaGNoS:

1. *Power Consumption*: This metric is computed as the total average power consumed in mW by SNs, within the M2WSNs, on their main components, i.e., CPU, low power module (LPM), and the transceiver operations (i.e., transmission and receiving), equation (5.4). The status of power consumption and resource utilization are periodically captured (e.g., every 10 secs) through the *Powertrace* module available in ContikiOS [108].
$$\overline{P}_{total}[mW] = P_{CPU} + P_{LPM} + P_{transceiver} \tag{5.4}$$

2. *Event Reporting Latency*: It is computed as the average latency from the time a working node detects an event and generates an event reporting packet to the time the first event packet is received at the sink node, equation (5.5). The emulation time of Cooja-ContikiOS was used to compute this metric.
$$\overline{\tau}_r[msec] = t_{recv-1st-event-pckt} - t_{event-detected} \tag{5.5}$$

5.3.2 Emulation Results

The results presented in this section are an average of the overall data collected on the set of emulations performed, with a 95% confidence interval for reliability on the measurements obtained [83].

Figure 5.10 shows the performance evaluation results of a small network (i.e., one grid with ten SNs and one sink node) using different multimodal switching mechanisms. The SNs within the network remains fixed during the whole emulations time. The rol of working node is cyclical rotated between SNs based on the node-queue built during the settle of the network. All mechanisms were implemented and evaluated under the same network parameters and topology conditions (Table 5.2 and Figure 5.7). The results are an average of 40 emulation runs per mechanism.

Hence, the results in Figure 5.10 indicate that M2-DaGNoS consumed $(1.89 \pm 5.4 \times 10^{-3})$ mW, performing better than HNS, $(57.40 \pm 1.3 \times 10^{-5})$ mW, and similar to eHNS, $(1.70 \pm 3.0 \times 10^{-3})$ mW, regarding power consumption, thanks to its multimodality behavior in the data gathering operations and the low-duty-cycling technique employed at the MAC layer (i.e., ContikiMAC at 8Hz). Regarding event reporting latency, HNS reports faster than the others, $(0.75 \pm 2.2 \times 10^{-4})$ ms, due to the always-on scheme implemented in all SNs in the network, but higher energy is required. M2-DaGNoS provides lower latency for reporting the first event packet, $(2.96 \pm 3.5 \times 10^{-1})$ ms, when compared to eHNS, $(4.20 \pm 6.3 \times 10^{-2})$ ms, thanks to the PED algorithm.

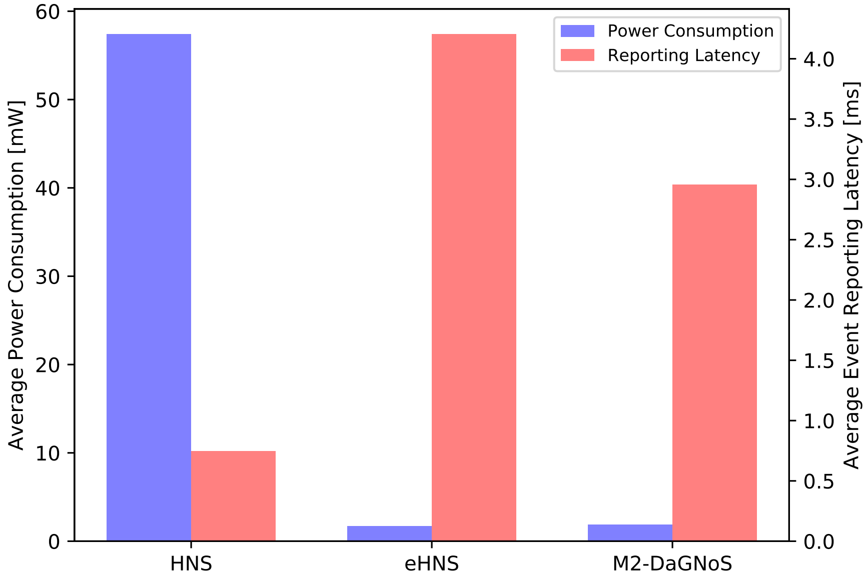


Fig. 5.10.: Performance evaluation of a small M2WSN using different multimodal switching mechanisms.

We perform a comparison between eHNS and M2-DaGNoS regarding their convergence, i.e., all nodes within a grid can build their own node queue **Q** and successfully execute the multimodal switching procedure during the whole emulation period. The HNS mechanism implements the same algorithms at the application layer than eHNS that is an enhanced version of HNS. Therefore, HNS presents convergence issues as eHNS, and we only consider M2-DaGNoS and eHNS mechanisms for the convergence study, given their close performance, especially regarding power consumption.

We execute 20 emulation runs with different seeds, under the same topology (see Figure 5.9a) and configuration parameters (see Table 5.2) for the convergence study. Figure 5.11 shows the convergence behavior of eHNS and M2-DaGNoS when the density of the M2WSNs increases, i.e., the number of grids, where each grid includes ten SNs. The results indicate that the convergence of eHNS worsens with the increase of the number of grids (i.e., nodes), while the M2-DaGNoS mechanism guarantees 100% of convergence. Figures 5.12a and 5.12b shows a example of a convergence study between M2-DaGNoS and eHNS mechanisms. The red stars (*) indicate that eHNS did not converge, i.e., the working node within a grid was not able to build the node queue **Q**. Hence, the multimodal technique is not executed.

The limitation of eHNS is due to the random values employed at the start time (e.g., T_{in}) defined before the node queue building process, where two or more SNs, that run the eHNS mechanism independently, have a high probability of setting the same T_{in}. Consequently, two

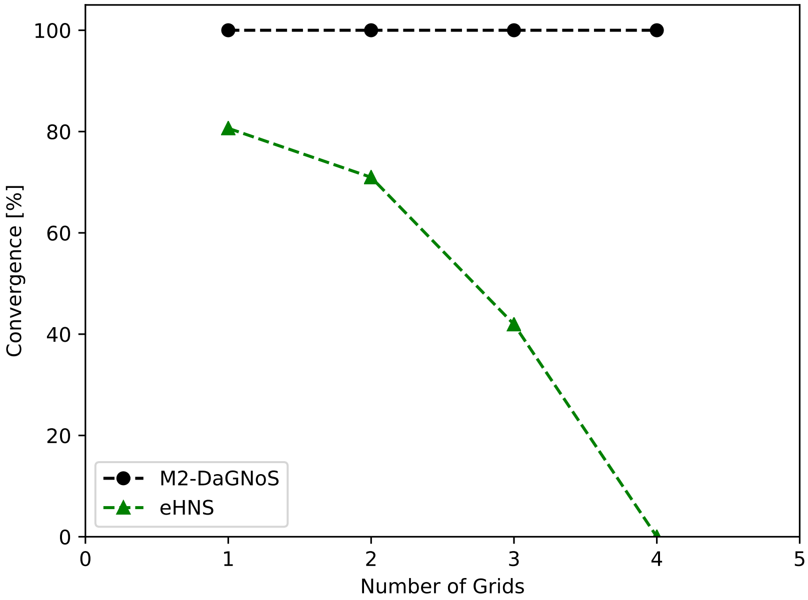
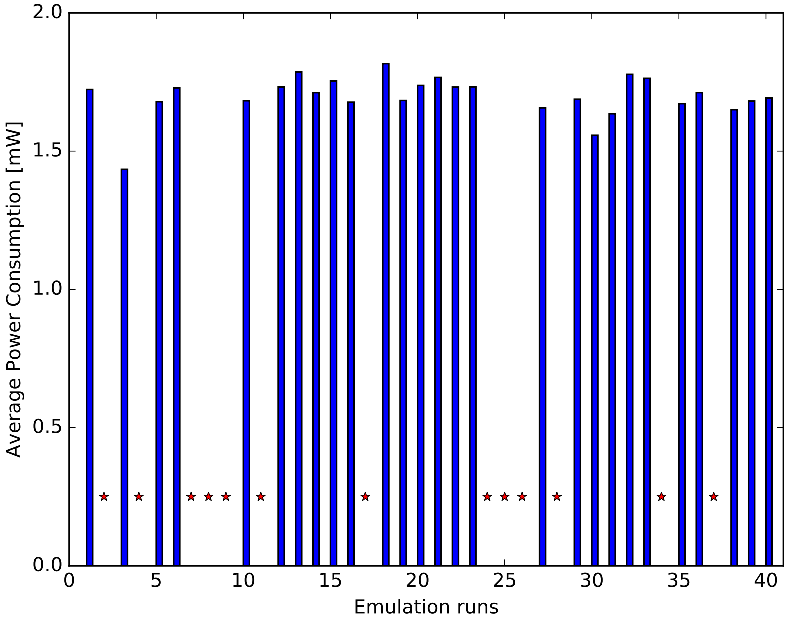


Fig. 5.11.: Convergence behavior of eHNS and M2-DaGNoS when the density of the M2WSNs increases, i.e., the number of grids, where each grid includes ten SNs.

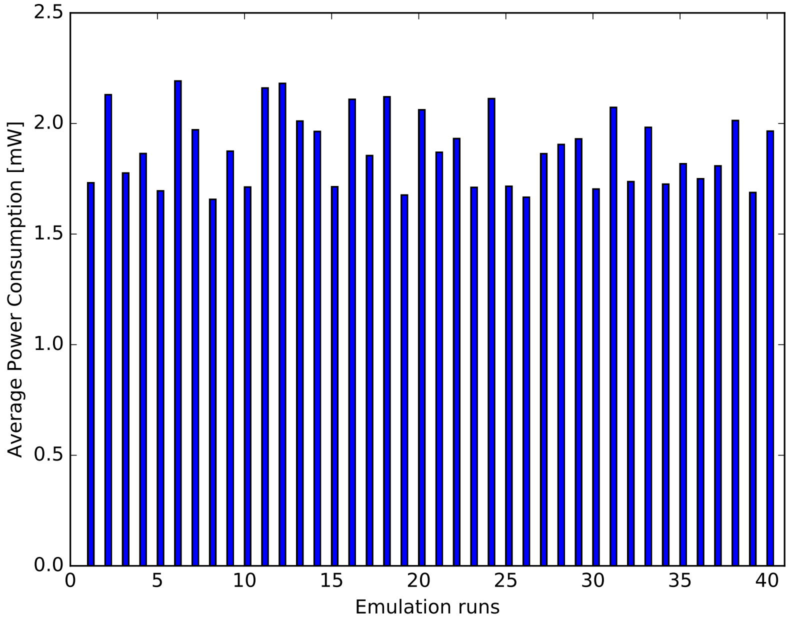
or more SNs are claimed as a working node at the same grid, causing that the mechanisms stuck there without able to continue to the next step. Therefore, we solved eHNS limitation in M2-DaGNoS by performing a neighbor discovery technique during the settlement of the network with the Node Scheduling procedure proposed in Algorithm 2. As a result, we obtain a neighbor table that serves as the node queue of every node in the grid and avoids the issue of simultaneous working nodes.

Figure 5.13a and 5.13b show the results of 20 evaluation runs under different number of adjacent grids (e.g., see Figure 5.1), regarding the average of event reporting latency and power consumption, where we compare M2-DaGNoS and eHNS mechanisms, under the same topology (see Figure 5.9a) and configuration parameters (see Table 5.2), as in Figure 5.10. We compute the metrics of eHNS using only those emulations that successfully run.

Within each grid, an event is generated independently based on the temperature profile presented in Figure 5.8. Consequently, there are moments in which more than one event coincides in the M2WSNs. Therefore, the sink should be able to attend several events within a period (worst case). The results indicate that the power consumption and the event reporting latency increase while the number of grids increments, presenting higher values for a more significant number of grids that is a typical behavior in WSNs applications. The abrupt increments, especially regarding the event reporting latency, might be due to interference between grids (share the same physical channel using CSMA/CA), re-transmissions, multi-

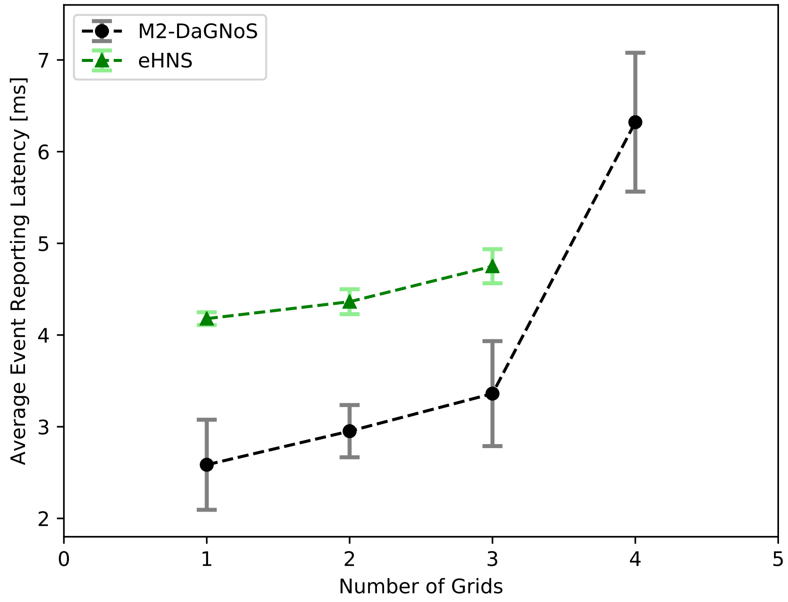


(a) eHNS mechanism.

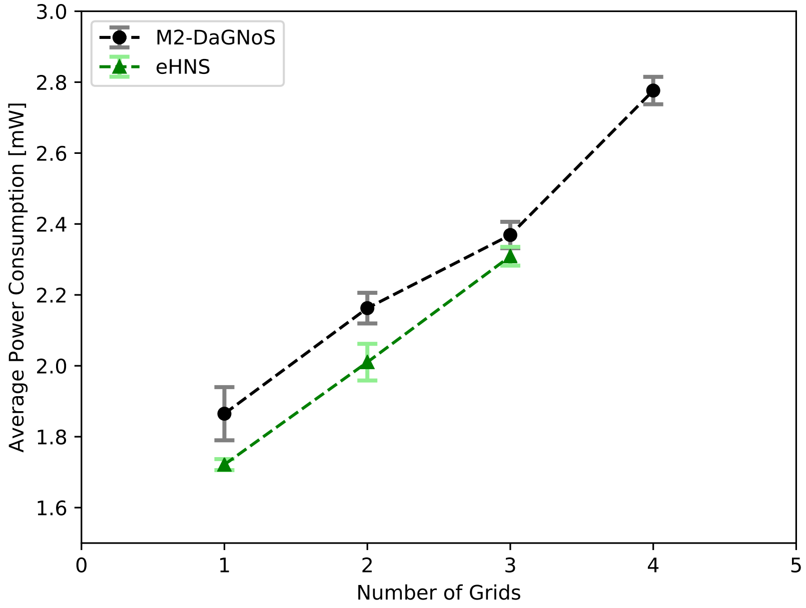


(b) M2-DaGNoS mechanism.

Fig. 5.12.: Comparison between eHNS and M2-DaGNoS regarding its convergence (i.e., every node builds its own node queue **Q** and successfully executes the multimodal switching procedure during the emulation period). The red stars (*) indicate that the mechanism did not converge.



(a) Average Event Reporting Latency.



(b) Average Power Consumption.

Fig. 5.13.: Performance evaluation results of M2-DaGNoS and eHNS under a different number of grids. Each grid has ten nodes, and each one runs the M2-DaGNoS and eHNS mechanisms independently. The event is detected in all grids at different times. No results are shown for eHNS in four grids because it did not converge.

hop communications, several bursts of packets attended by the sink during an emergency, generating extra latency into the receiving buffer of packets. Finally, the results indicate that eHNS performs better than M2-DaGNoS regarding the power consumption when the eHNS converges, but eHNS presents a very low or null reliability (convergence) performance for dense M2WSNs when compared to M2-DaGNoS. No results are shown for eHNS in four grids because it did not converge in any of the emulation runs executed (in average 85 tries), as shown in Figure 5.11.

M2-DaGNoS can be combined with an WuRx solution (an emerging solution for WSNs applications) to further manage the trade-off of event reporting latency for energy efficiency in M2WSNs [83]. Besides, considering multiple sinks into the M2WSNs might help to reduce the packet overload during short emergency periods when a single sink is used. Besides, the higher power consumption of those nodes nearby the sink (known as the energy hole problem [109]), and the higher event reporting latency for those far away could be reduced (see Figures 5.9a and 5.9b). The latter implicates the need of efficient clustering and power-aware routing mechanisms to support a scheme of multiple sinks [110, 111, 112].

5.4 M2-DaGNoS Validation

In this section, we present the setup used for the M2-DaGNoS validation and the corresponding results and analysis.

5.4.1 Experimental Configuration

We configured a small-scale M2WSNs in a laboratory environment to evaluate and validate the performance of the M2-DaGNoS mechanism under real motes. The M2WSNs includes eleven FeuerWhere motes (FW-nodes) [69] running ContikiOS and deployed on a well-defined topology, where the sink node, with a permanent power source, is located at one corner as shown in Figure 5.14. The sink node is connected to a PC for data logging. Motes were separated from each other 25cm to 50cm, due to space limitation in the moment of performing the experiments. A summary of the main features of FW-node is provided in Table 5.3. The M2-DaGNoS's configuration parameters are the same used in the emulation study (see Table 5.2).

We defined and performed two experimental setup scenarios to evaluated the operation of M2-DaGNoS in hardware, as shown in Figure 5.15. The first scenario considers only a normal circumstance (CMnt mode only), i.e., without the occurrence of an event, where nodes within the newtork cyclical rotate the rol of working node during the experiment. The working node continuously reports data packets of the supervising area to the sink,

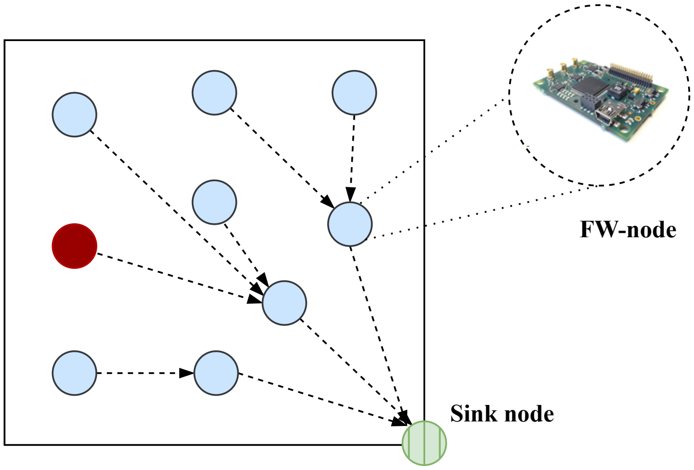


Fig. 5.14.: A small-scale network implementation using FeuerWhere nodes for M2-DaGNoS test evaluation. The red dot indicates the working node and the blue dots, the assisting node. All nodes implemented using FW-node.

Tab. 5.3.: Summary of the main features of the FeuerWhere node (Based on [69]).

Feature	FW-node
Processor	16-bit TI MSP430F5438A
Memory	16KB (RAM), 256KB (Flash), 4MB (external Flash)
RF technology	CC2520 (IEEE 802.15.4 2.4GHz), CC1101 (European 868MHz), CC2500 (2.4GHz)
Sensor	External: Temperature, Relative humidity
Power Specifications	Bateries, 0.9-6.5 V DC
Software	ContikiOS, LangOS

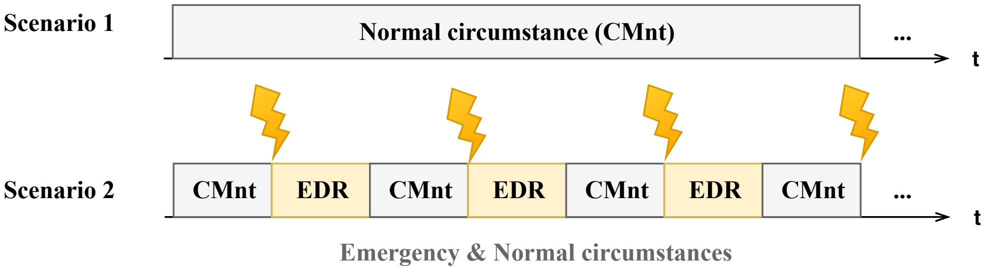


Fig. 5.15.: Experimental setup scenarios.

every $T_{reporting} = 10secs$, without an interaction with its assisting nodes. The assisting nodes remain in an sleeping state during the operation of the newtork. In the second scenario, normal circumstances, and emergencies are considered during the whole operation of the network. The events are introduced based on the temperature profile defined for the emulation study (Figure 5.8). In the second scenario, we evaluate in hardware the "multimodality" of M2-DaGNoS regarding the data-reporting mode. The current working node switches between CMnt and EDR modes based on the circumstances, i.e., in the presence or absence of an event, as shown in Figure 5.15. During an emergency, the working node reports event data packet every 5 seconds, using the EDR mode. Besides, it is assisted by the redudant nodes of the network. These assisting nodes report more data packets associated with the event toward the sink for tracking purporses.

In summary, for the defined scenarios, it has been configured a packet rate of 1 packet/5secs, during an emergency; while, 1 packet/10secs during the normal circumstance. Hence, a node generates on average 9 data packets during its working period and 2 additional data packets per neighbor, during the assistance period (nodes were located very close from each other. Nine neighbors in total per node). Each node performs the role of the working node, on average, 7 times for 60 minutes. Therefore, on average, each node introduces into the network 189 data packets for 60 minutes of one experiment trail–without considering packet retransmissions. The payload for every data packet is 37 bytes.

The same performance metrics used during the emulation study has been considered for the M2-DaGNoS operation validation in hardware. The overall data collected during two experiments of 60 minutes each, one per scenario, have been employed to compute the average power consumption of the network with a 95% interval confidence. For the second scenario, on average, a total of 1890 packets have been generated, and 360 packets, for the first scenario. Regarding the event reporting latency, it has been computed by considering the same physical node when it performs the working node role and reports the first event data packet to the sink. Two distance hops to the sink have been considered: 1-hop and 2-hops–include the extreme nodes, i.e., source and sink nodes, having one intermediate node between them. The results are an average of the data collected during 25 working periods (i.e., when the mote performs the working node role). The 95% interval confidence is indicated.

We measured the power consumption of a FW-node and the whole network using a Wireless Debugging and Power Measurement System proposed in [113]. For the event reporting latency, we measured the difference of two trigger signals generated by the working node (from the moment the event packet is generated) and the sink node (from the time the event packet is processed), respectively, using a digital oscilloscope.

5.4.2 Validation Results

The current consumption profile of one mote within the M2WSNs, supporting M2-DaGNoS mechanism under a single-radio architecture, is shown in Figure 5.16. The profile indicates the different states of M2-DaGNoS mechanism according to the state machine diagram proposed in Figure 5.4. The higher current peaks are due to the packet transmission and reception operations (e.g., data and event packets (continuous reporting and sampling states, respectively), working and assisting messages (working and assisting states, respectively)).

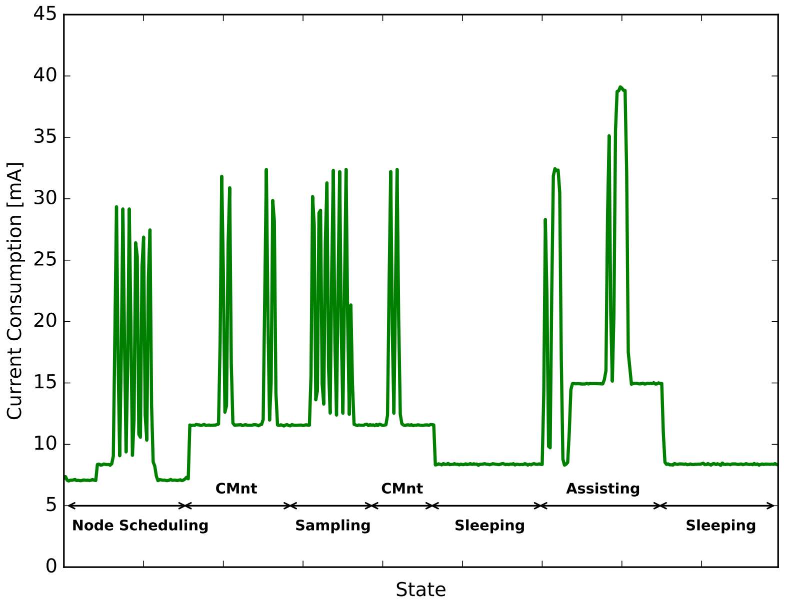


Fig. 5.16.: Current consumption profile of one mote within the M2WSNs when supporting the M2-DaGNoS mechanism at its different states under a single-radio architecture.

Tab. 5.4.: M2-DaGNoS performance evaluation results (real motes).

Scenario	Avg. Power Consumption [mW] *Whole M2WSNs*	Avg. Event Reporting Latency [ms] *1-hop*	 *2-hops*
First	$392.5 \pm 26.2 \times 10^{-3}$	$N.A$	$N.A$
Second	$478.6 \pm 53.1 \times 10^{-3}$	136.3 ± 6.0	430.2 ± 11.7

Finally, during the Assisting state, the base-line for the current consumption is higher than during the working state (CMnt or Sampling), in the order of 3 mA that can be due to several causes such as idle-listening and extra timers that the MCU should attend.

The performance results of M2-DaGNoS in hardware is shown in Table 5.4 regarding power consumption and event reporting latency–for different distance hops to the sink. The results indicate that M2-DaGNoS consumes slightly more energy when emergency and normal circumstances are presented simultaneously within the M2WSNs. In this scenario, the power consumption of the whole M2WSNs is significantly higher due to the assistant nodes operations that attend the working node during an emergency when compared to a single data reporting (e.g., the first scenario in Figure 5.15, where only one mote, the working node, is available during the entire continuous monitoring operation).

Regarding the event reporting latency, the results show that the further the mote is from the sink, the higher is the latency (end-to-end delay). This behavior is typical in multi-hop communications. However, the difference between 1-hop to 2-hops is more than three times. This difference can be due to many causes. Under an LDC scheme, one node checks the channel for radio activities every 125ms, giving the channel check rate configured in ContikiMAC (8Hz, default value). Consequently, the source node should wait up to 125ms to forward a packet under ideal conditions. Under the effect of a real channel (over the packet transmission, e.g., collisions, distance losses, multipath propagation), the waiting time is increased, e.g., 136.3ms for 1-hop. In a multihop communication, the latency is even bigger, giving the contribution that represents one more hop to sink and even more, the packet loss generated when it is sent toward the sink over a real channel [83].

Therefore, there is one more reason to consider a wake-up protocol based on the WuRx paradigm that promises to overcome latency issues during the process of transmitting a packet toward the sink [83]. In this sense, we consider that the M2-DaGNoS mechanism over a radio-duty-cycling technique based on WuRx is promising, regarding event reporting latency, as described in the next chapter.

5.5 Remarks

In summary, this chapter provides the design, implementation, and evaluation of M2-DaGNoS, an enhanced MultiModal Switching mechanism for Data Gathering and Node Scheduling that is suitable for monitoring applications in M2WSNs under normal circumstances and emergencies. The implementation has been done in ContikiOS, an open-source operating system for the IoT solutions. The performance of M2-DaGNoS is evaluated against the state-of-the-art multimodal switching mechanisms implemented under a single-radio architecture, using the Cooja emulator through an extensive simulation study. Finally, the M2-DaGNoS mechanism is validated in a laboratory-based environment regarding power consumption and event reporting latency under different experiments.

When comparing the emulation study results and the results obtained using real motes regarding power consumption and reporting accuracy (only for the M2-DaGNoS mechanism evaluation), they highly differ. The testbed is implemented in a laboratory-based environment where there are several sources of interference that might affect M2WSNs communications such as devices transmitting at 2.4GHz, path-loss and multipath propagation, power supplies, and obstacles (furniture), i.e., the effects of a real wireless communication channel; also, defects in mote hardware, that increase the metrics under study. In contrast, strong assumptions are made in the emulation study such as to employ an “Unit Disk Graph Medium (UDGM): Distance Loss” radio medium that does not consider all the effects of a real communication medium (e.g., mutilpath propagation, collisions). Therefore, the testbed validation is crucial to be performed when a new framework or technique is proposed, a fact that has not been considered in several works proposed in the literature. However, the emulations study allowed us to get insights about the behavior of the M2-DaGNoS operation under an LDC scheme and different settings parameters (e.g., number of nodes within the M2WSNs) when compared against similar techniques proposed in the literature, such the eHNS mechanism.

Finally, we have assumed that an M2WSN is divided into several clusters (grids) by a previous clustering procedure. For example, by executing a LEACH[7]-based clustering protocol [20]. Besides, we considered for the M2-DaGNoS design and evaluation, those grids near (adjacent) to the sink. However, non-adjacent grids can be formed in large-scale M2WSNs, case not considered in this work. As future work, we propose to implement an algorithm that allows joining adjacent and non-adjacent grids into an extended minimum spanning tree for data-reporting toward the sink, and separately, each grid performed the corresponding messages exchanging for M2-DaGNoS operation via a collection tree. Hence, from the data dissemination perspective, the M2WSNs can be organized into two-logical-overlapping layers, one for data communication between adjacent and non-adjacent grids and another for messages exchanging within each grid.

In the first logical layer, adjacent and non-adjacent grids can be joined and maintained by implementing algorithms such as the Dynamic Gallager-Humblet-Spira (D-GHS) algorithm for WSNs [105]. D-GHS builds and maintains a minimum spanning tree, where nodes of the network are initially fragmented, i.e., nodes belong to different fragment–a connected subgraph with bi-directional edges. During the tree construction, D-GHS subsequently merges fragments into a new one until it remains only one main fragment, the minimum spanning tree of the graph. Once the minimum spanning tree is constructed, the sink becomes the root of the tree. This is done by transmitting an initiate message over the sink's branch edges. After receiving an initiate message, every node in the tree selects a new parent that leads to the sink. Hence, data dissemination can be performed toward the sink. Finally, in the second logical layer, a collection tree protocol can be executed to exchange M2-DaGNoS messages within each grid. Different logical communication channels can be set for messages filtering and exchanging within each logical layer.

---

[7] Low Energy Adaptive Clustering Hierarchy (LEACH).

Framework Performance Evaluation

6

In this chapter, we present the performance evaluation of the proposed framework for M2WSNs based on the WuR paradigm and cross-layer design, introduced in Chapter 3. The framework is implemented in ContikiOS, from the application layer to the physical layer, and its performance is evaluated in a laboratory-based environment using real motes. Finally, we compare the proposed framework against a single-radio architecture with a low-duty-cycling technique introduced in Chapter 5.

6.1 Overview

In previous chapters, we mentioned that M2WSNs are a solution proposed for those monitoring-oriented applications where it is necessary to simultaneously manage normal circumstances and emergencies via a reliable and energy-efficient multi-hop communication. Like traditional WSNs, the M2WSNs include several SNs with limited energy and computational resources that usually are randomly deployed in an area of interest to monitor physical variables (e.g., temperature) and work collaboratively to detect and track the occurrence of an event (e.g., fire forest). Compared to traditional WSNs, SNs within M2WSNs are characterized by a multimodality feature regarding their data-gathering scheme and radio architecture.

The SNs of M2WSNs execute a multimodal switching mechanism which grants them with the capability of immediately react upon the occurrence of an emergency, i.e., an event, using an event-driven data-gathering scheme to report to an observer the event and its associated data. After the emergency, during the period of calm of the event, this mechanism allows SNs to switch to a continuous or time-driven data-gathering mode to periodically emit up-to-date data of the current status of the supervising area.

To further improve the performance of M2WSNs, i.e., better energy-savings and at the same time, to overcome the higher data latency due to collisions and the “waiting period” presented in a traditional LDC protocol, the implementation of *Wake-up Radios* is considered. These radios have the capabilities to continuously monitor the wireless channel, allowing to reduce the data latency while consuming a small amount of energy when compared to those radios commonly employed in WSNs [94]. However, it is not enough to implement a dual-radio architecture at the physical layer, i.e., a WuR receiver and the main radio (for data transmission and reception), the network and MAC layers should be modified to support the advantages of the WuR paradigm.

Piyare *et al.* in [17] mentioned that it is missing a unified system and networking architecture under the WuR approach for WSNs, where applications can be implemented, without relying on simulation tools, but on real implementations or testbed. In recent years, some approaches have been proposed in the literature regarding this claim. The same authors proposed in [92]–KRATOS, an open-source hardware-software platform for wireless networks based on long-range radio technologies such as LoRa and short-range WuR, running on the Contiki operating system, for testing and development of LoRa networks. This platform was used in [94] to validate a network architecture and on-demand TDMA MAC protocol oriented to achieve energy-efficient and responsive communication using LoRa under a receiver-initiated system, where the gateway has full control of the network in continuous monitoring-oriented applications.

Kumberg *et al.* in [22] proposed T-ROME, an simple energy-efficient cross-layer network protocol for WSNs based on the WuR paradigm that allows to use different transmission ranges on the main radio and WuR, and to optimize the relaying process by skipping hops when the sink is not available in one-hop communication, to save energy during the data dissemination that it is executed in a distributed manner and under a sender-initiated communication approach. T-ROME supports functions in the network and link layers, with non-cross-layer interactions between the application and physical layers.

Recently, Sutton *et al.* in [114] proposed BLITZ, a communication architecture for efficient event-triggered multi-hop WSNs that simultaneously supports low latency and energy-efficiency. To that end, BLITZ employed an interference-based network flooding approach for waking-up all nodes in the network in an asynchronous way and a synchronous and topology-agnostic protocol for data dissemination between the source and the host, following a sender-initiated communication approach. The data dissemination procedure is managed by the host. To mitigate the false wake-ups that could arise during the wake-up procedure due to interference, the authors implemented a distributed wake-up classifier using a decision-tree technique instead of an addressable or id-based wake-up scheme.

In Table 6.1, we present a qualitative comparison between our proposal and the closest work aforementioned. All proposals employed a WuR approach in the design, some following an id-based scheme (i.e., the WuRx supports addressing) or a wake-up classifier and others, a broadcast-based wake-up scheme. Different networking techniques were implemented at the network and link layers. Our framework implements an adapted version of the Collection Tree Protocol proposed in the RIME stack of ContikiOS based on the WuR paradigm when compared to the work in [94] that employed the original version available in RIME implemented for LoRa receiver-initiated communications. We considered a fully distributed

Tab. 6.1.: Qualitative comparison with similar work.

Authors	Reporting Scheme	Networking Techniques	Wireless Technology	WuRx Scheme	Communication approach	Decision-making approach
Kumberg *et al.* [22]	Time-driven	T-ROME	IEEE 802.15.4	Id-based	Sender-Initiated	Distributed
Piyare *et al.* [94]	Time-driven	RIME/On-Demand TDMA	LoRa	Id-based & Broadcast-based	Receiver-Initiated	Centralized
Sutton *et al.*[114]	Event-driven	Interference-based Flooding	IEEE 802.15.4	Classifier	Sender-Initiated	Semi-Centralized
This work	**MultiModal**	2R-RIME/CSMA	IEEE 802.15.4	Broadcast-based	Sender-Initiated	Distributed

and cross-layer approach for decision-making regarding the wake-up procedure and the data-reporting mode, when compared to the others proposals that allow SNs to make decisions, without relying on their neighbors and the sink, only during the wake-up phase.

To our knowledge, our framework is the first to consider a *multimodal* approach regarding the data-reporting scheme combined with a dual-radio and networking architectures based on the WuR paradigm following a cross-layer design, where all layers of a traditional WSNs communication stack participate in the cross-layer interaction, from the application to the physical layers and vice-versa, making it suitable for reliable and energy-efficient multi-hop M2WSNs.

In this chapter, we present the performance evaluation results of the framework for M2WSNs based on the WuR paradigm introduced in Chapter 3. To that end, we conducted several experiments using real motes in a laboratory-based environment and compare the proposed framework against the single-radio architecture based on a low-duty-cycling technique introduced in Chapter 5. In the following sections, we provide more details of the proposed framework. Then, we describe the experimental configuration used in the testbed. Finally, the performance evaluation results are presented and analyzed. In summary, the main contribution of this chapter read as follows: performance evaluation and comparison between the proposed framework and traditional single-radio architecture via laboratory-based experiments.

6.2 Dual-radio Architecture Overview

As mentioned in Chapter 3, we propose a framework that follows a dual-radio architecture that consists of two independent radio modules, one main radio, for data transmission and reception procedures under the IEEE 802.15.4, and a secondary radio, for wake-up signals

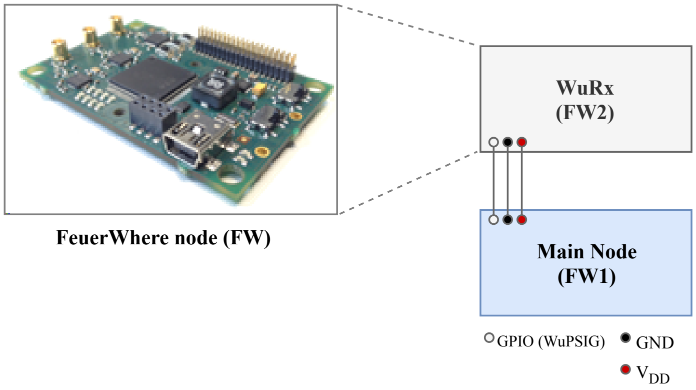


Fig. 6.1.: Example of a dual-radio hardware architecture using two FW-nodes from IHP: One as WuRx (always-on scheme) and the other, as a main node (in sleep mode).

reception. From the hardware perspective, the architecture is arranged as two side-by-side motes (e.g., FW-nodes [69]) connected via General-Purpose Input/Output (GPIO) ports, as shown in Figure 6.1, working at the same frequency band (e.g., 2.4GHz), i.e., an in-band channel scheme is assumed, and with independent power supplies. One FW-node (FW2) emulates a WuRx that main functions are: (1) Be always listening to the communication channel for any WuS coming from another sensor node in the M2WSNs; (2) If a WuS is received, the FW2 triggers an external interrupt signal (WuPSIG) from a GPIO to wake a second FW-node up, i.e., the FW1. The FW1 works as the primary node that is in a low power mode (LPM) or sleep mode until some interrupt signal actives it via GPIO. The main node has the following functions: (1) Receive a data packet from other sensor nodes in the M2WSNs after being woken up by the WuPSIG; (2) Process or relay the data packet towards the sink node and go back to its sleep mode. From now on, the first and second FW-nodes are named as WuRx and MR, respectively.

Both motes run a firmware under ContikiOS with the communication stack shows in Figure 6.2 that follows the structure of the proposed framework. At the physical layer, a hardware interface is added to the ContikiOS for wake-up receiver interrupts handler (i.e., between the WuRx and the CPU of the second FW using I/O ports (for WuPSIG)). A lower-power RF chip drivers (e.g., cc2520) with IEEE 802.15.4 (at 2.4GHz) is employed for data packet transmission and receiving procedures via the primary radio. At the MAC layer, the asynchronous 2R-MAC protocol is implemented as the Radio Duty-cycling (RDC) protocol. The MR supports CSMA/CA and 2R-MAC techniques at the link layer, the dual-radio collect tree protocol–2R-CTP, adapted from the RIME stack–, at the network layer (as described in Section 3.2.2), and the MultiModal Switching mechanism, M2DaGNoS, proposed in Chapter 5, at the application layer. The WuRx supports only a simplified version of 2R-MAC at the MAC layer for handling the WuS receiving and the WuPSIG triggering processes. The WuRx is configured in promiscuous mode, i.e., the radio can pass all the traffic generated in the channel to the CPU.

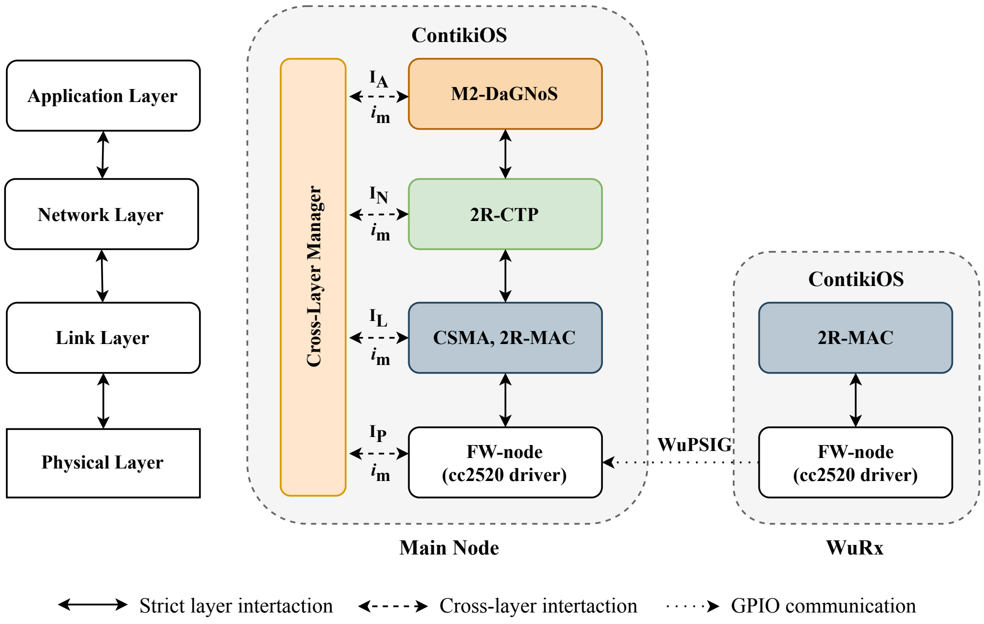


Fig. 6.2.: Example of a dual-radio software architecture from a communication stack perspective.

As mentioned in Section 3.3, the CLM module handles the non-adjacent interactions between the modules included in the framework (see Figure 6.2–the M2WSNs framework from the software perspective). The current implemented version of the framework supports the cross-layer interaction of link to network layers, network to application layers, application to physical layers and vice-versa, i.e., i_m provides information about the wake-up period (i_{L_1}) for packet re-transmissions purposes, re-transmissions status (i_{L_2}) for sending timeout adjustments, sending status (i_{N_1}) for working messages management, the neighbor table (i_{N_2}) for the node queue building procedure, the desired (i_{A_3}) and current (i_{P_1}) power TX. As future work, it remains to implement the cross-layer interaction of application to network, and application to link layers that might add additional features to the framework such as priority sending packet management during emergencies, dead nodes management and switching between different RDC schemes, e.g., LDC and WuRx.

A Dual-Radio Duty-Cycling. 2R-MAC is proposed to manage the switching on and off procedure of the radio transceiver, i.e., the radio should keep off as much as possible to achieve better energy-saving results, but at the same time, the radio should be able to receive an incoming packet. The 2R-MAC technique is an RDC protocol that works under a dual-radio architecture based on a WuRx. It allows the coordination between the main radio and the WuRx and offers an interface to the higher layers protocols as CSMA/CA and 2R-CTP via cross-layer interactions.

The 2R-MAC technique is a modified version of the W-MAC protocol proposed in [82]. The W-MAC has been developed for the sky-mote (CC2420) as an emulator plugging for

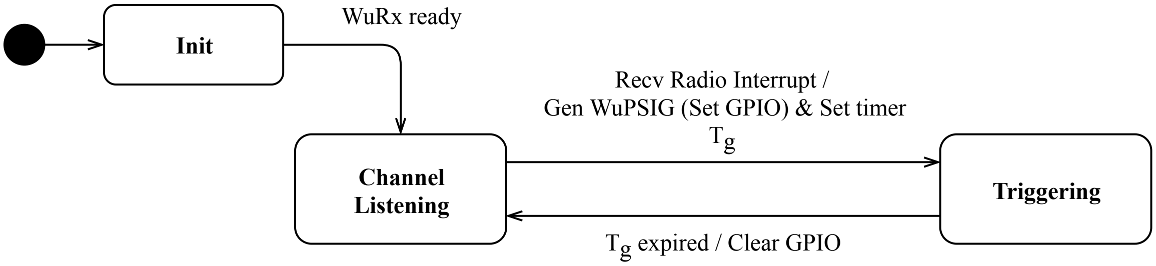


Fig. 6.3.: WuRx application state machine diagram using 2R-MAC protocol.

Cooja. W-MAC operates on in-band channels (at 2.4GHz) and ID-based schemes (i.e., with addressing support), and it is compatible with the CSMA and RIME stack. Therefore, multi-hop communications are supported. Besides, 2R-MAC operates similar to W-MAC but under a broadcast-based wake-up scheme, i.e., non-addressing, implemented based on [92].

From the WuRx side, the 2R-MAC protocol is only listening to the channel, awaiting to any data packet (i.e., WuS) to arrive and then, to wake the MR up. The WuRx application that also provides a hardware interface between the WuRx and the MCU of the main node, follows the state machine presented in Figure 6.3. After settling the WuRx, the application transits to the *Channel Listening* state, and it is ready to receive any WuS. Upon a radio interrupt, i.e., a WuS has been received, the WuRx passes to the *Triggering* state and triggers a WuPSIG signal (i.e., a high pulse) via a GPIO port. The WuPSIG remains active during a T_g period (i.e., a timer is set to some milliseconds). When the timer expired, the GPIO is cleared, and the WuRx returns to the *Channel Listening* state.

From the MR side, the 2R-MAC protocol follows a sender-initiated scheme, i.e., “a message source triggers the receiver to wake-up” [82]. Based on this scheme, the MR follows the state machine shown in Figure 6.4. After settling the MR, the radio is turned off, and the main node transits to the *Sleeping* state. An Interrupt Service Routine (ISR) is configured to handle the WuPSIG interrupt generated by the WuRx. Upon the interrupt signal, the radio is turned on (put in receiving mode), and the application remains in the *Receiving* state during a T_{AW} period. If no data packet is received during this period, the node goes back to the *Sleeping* state. Otherwise, the data packet is received, then processed to be passed to the higher layers. The T_{AW} has been set to 26 clock ticks (i.e., 204ms approx.)–sufficient time for receiving a packet from other nodes. The T_{AW} parameter is shared, via the CLM module, to the RIME stack to set the retransmission packet time for transmission purposes. The application transits from the *Sleeping* to the *Transmitting* state when the higher layer has a data packet ready to transmit over the channel.

When an event occurs within the M2WSNs, the SNs follows the procedure presented in Figure 4.1, and shown again in Figure 6.5 that describes a dual-radio multi-hop communication on a tree-like topology, built and maintained by 2R-CTP. The right sketch shows a multi-hop

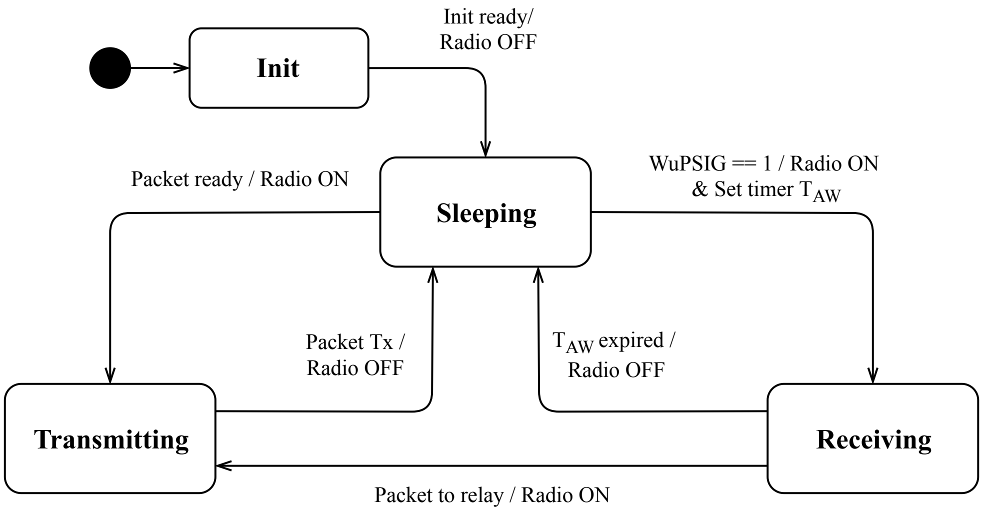


Fig. 6.4.: Main node application state machine diagram using 2R-MAC protocol.

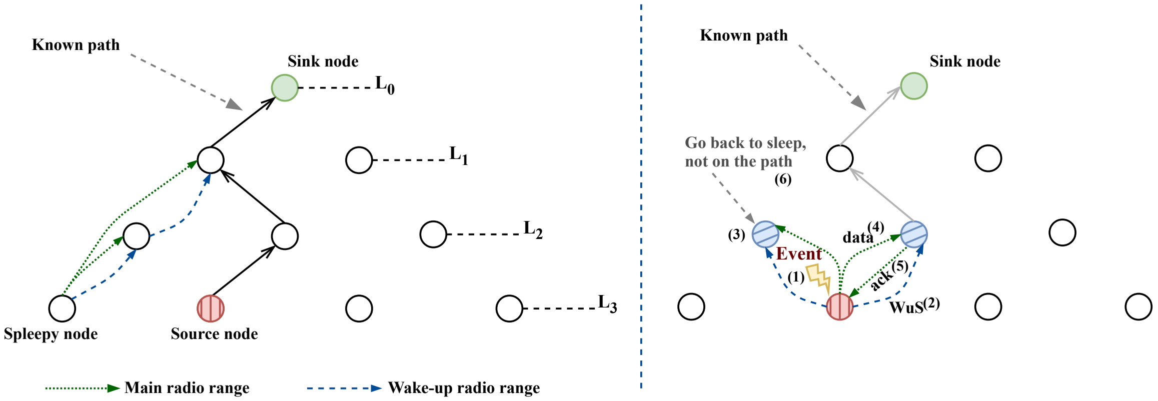
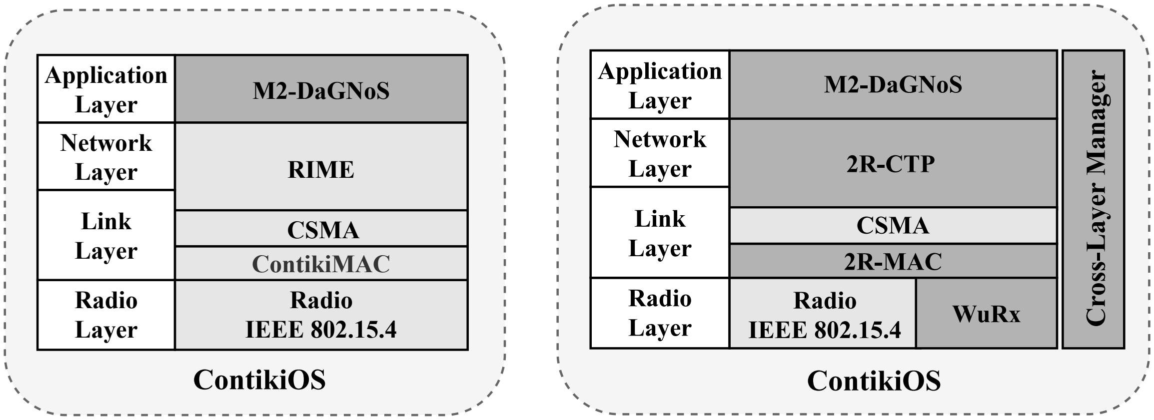


Fig. 6.5.: A dual radio multi-hop communication schematic on a tree-like topology.

operation of the wake-up protocol proposed: (1) A source node detects an event, reports and propagates it through a known and reliable multi-hop routing path towards the sink, previously defined by 2R-CTP. (2) The wake-up protocol operates under a transmitter-initiator scheme, where the source starts the communication by first sending a WuS packet (the same event data packet) using its main radio. (3) The WuS packet wakes up all potential receivers, (i.e., child and parents) within the WuR range (i.e., those SNs that have a WuRx integrated). (4) After sending the WuS, the source waits for a short time, and then emits an event data packet. (5) Afterward, the source waits for an acknowledge packet (ack). If no ack is received during a predefined time, the sender transmits the same data packet again, until an ack arrives or the number of retransmissions is exhausted. (6) The non-destination SNs turn on their main radio after receiving the WuS, remain active until a data packet is received, but then go back to sleep because the destination address does not match their address. Finally, this procedure is repeated in each hop within the multi-hop path until the sink receives the event data packet.



(a) Single-radio architecture.
(b) Dual-radio architecture.

Fig. 6.6.: An overview of communication stacks implemented under different radio architectures. The dark gray modules are proposed for the framework.

The procedure described before is the “worst" scenario for wake-up and data dissemination toward the sink due to: (i) non-addressing support is used (i.e., all SNs within the neighborhood are woken up to receive a packet; (ii) every node in the path should send at least two data packets, one for wake-up the next-hop and the other for data transmission; and finally, (iii) every next hop in the known path should acknowledge the data packet. We decided to implement the “worst case" to compare our approach against a traditional low-duty-cycling technique under a single-radio architecture, giving that if our framework performs better against the traditional proposal, we might expect to get further improvements with other approaches [114, 83, 22], regarding event reporting latency and energy savings.

6.3 Framework Validation

In this section, we present the setup used for the framework validation and the corresponding results and analysis.

6.3.1 Experimental Configuration

In the previous section, we presented the dual-radio architecture and the 2R-MAC protocol based on WuRx implemented in the proposed framework. The entire framework has been implemented in ContikiOS, i.e., from the application layer to the physical layer and tested in hardware.

In the subsequent sections, we compare our framework against a single-radio architecture, as shown in Figure 6.6a. The single-radio architecture employs only one radio with IEEE 802.15.4 support when compared to our framework. As shown in Figure 6.6b, our framework

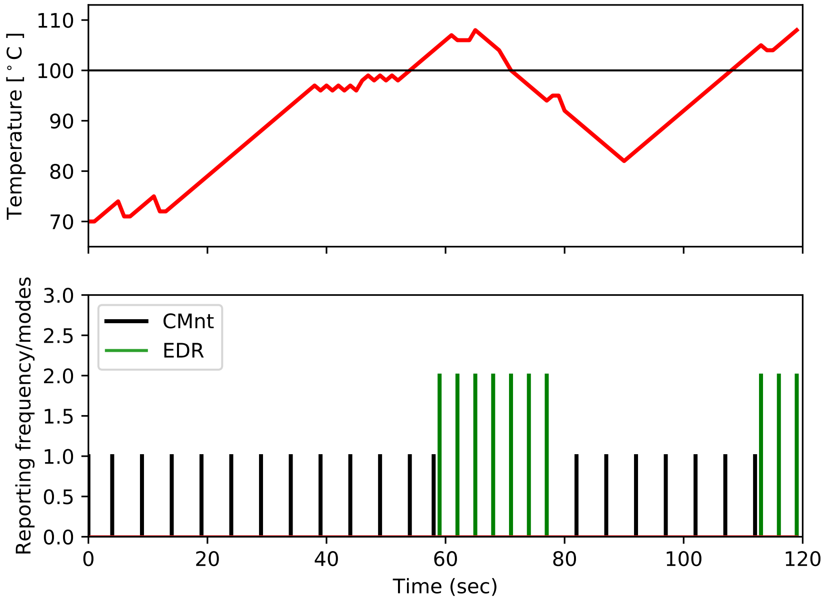


Fig. 6.7.: Temperature profile with the reporting modes and frequencies for the validation process.

uses two radio modules: one main radio for data transmission and reception procedures under the IEEE 802.15.4, and a secondary radio, for wake-up signals reception, tunned to the frequency operation of the primary radio (e.g., 2.4GHz). At the link layer, the single-radio architecture implements ContikiMAC [50]–the traditional asynchronous low-duty-cycling mechanism widely evaluated in the literature [51, 115, 83]–, and our framework implements 2R-MAC technique–described in section 6.2. Both architectures use the CSMA/CA technique, as the medium access control strategy.

At the network layer, the single-radio architecture utilizes the original CTP available in the RIME stack compared to our framework that implements the 2R-CTP with RIME communication functions. Both architectures repeatedly send the full data packet until the receiver acknowledges the packet. Finally, at the application layer, the M2-DaGNoS mechanism proposed in Chapter 5 is implemented in both proposals. The sink node is set to be always-on during the whole operation of the network. Hence, a null RDC is implemented in our framework, and for the single-radio architecture, ContikiMAC configured in its active mode.

For the performance evaluation comparison, we choose a scenario that considers both normal circumstances and emergencies, as shown in the second scenario of Figure 5.15. Under a normal circumstance, the working node is reporting to the sink every 5 secs (black lines), and during an emergency, every 2.5 seconds (green lines) within 120 seconds ($T_{gathering}$). The data-reporting frequency ($T_{reporting}$) is smaller than the one used in the emulation study

Tab. 6.2.: Testbed setting parameters.

Parameter	**Dual-Radio**	Single-Radio
Wake-up time, T_{AW} (Main radio)	**Fixed, 204ms approx.**	Dynamic
MAC Layer	**CSMA (Contiki version) 2R-MAC**	CSMA (Contiki version) ContikiMAC (8Hz)
Network layer	**2R-CTP with RIME**	RIME
Application layer	**M2-DaGNoS**	M2-DaGNoS
Max. retransmissions (Network layer)	**5**	5
Max. CSMA retransmissions	**1**	2 (default)
Packet rate (Data)	**1 packet/5 seconds**	1 packet/5 seconds
Packet rate (Event)	**1 packet/2.5 seconds**	1 packet/2.5 seconds
Payload (Data) (D, E)	**37 bytes**	37 bytes
Payload (Ack)	**5 bytes**	5 bytes
Distance hops	**3**	3
Main node	**FW-node**	FW-node
WuRx	**FW-node**	N.A.
Sink radio duty cycling	**Always-on (nullrdc)**	Always-on (ContikiMAC)

(10secs [25]). Likewise, the working period ($T_{gathering}$) is much longer than the 50secs [25] used during the emulation study. Events are introduced into the network according to the temperature profile shown in Figure 6.7. The number of event occurrences is the same as the one used in the emulation study (2 occurrences).

By employing the profile presented in Figure 6.7, a longer working period and a shorted data-reporting frequency, the working node can report data at least 20 times during the CMnt, and 10 times, during an emergency using the EDR capabilities, when compared to the profile proposed in Figure 5.8, 4 and 5 times, respectively. Hence, we increase the amount of packets introduced into the network and the amount data to compute the performance metrics that allow us to make a comparison between our framework and the single-radio architecture.

In summary, for the defined scenario, it has been configured a packet rate of 1 packet/2.5secs, during an emergency; while, 1 packet/5secs during the normal circumstance. Hence, a node generates on average 30 data packets during its working period and 2 additional data packets per neighbor, during the assistance period. Therefore, on average, each node introduces into the network 224 data packets for 60 minutes of one experiment trial–without considering packet retransmissions. The payload for every data packet is 37 bytes. On Table 6.2, the most important parameters configured in the testbed are summarized.

We define three quantitative metrics to compare the performance of our framework against the single-radio architecture: (1) *Power Consumption* as the total average power consumed in mW by the source node within the M2WSNs on its main components: CPU, transceiver

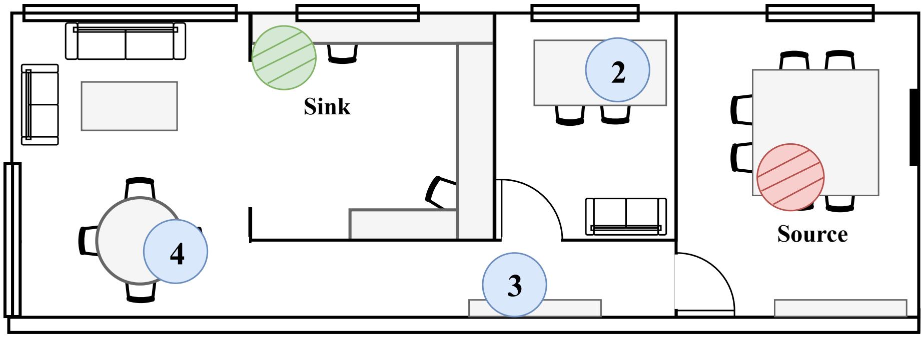


Fig. 6.8.: Map of the indoor testbed setup (area of 93m^2) used for the proof-of-concept.

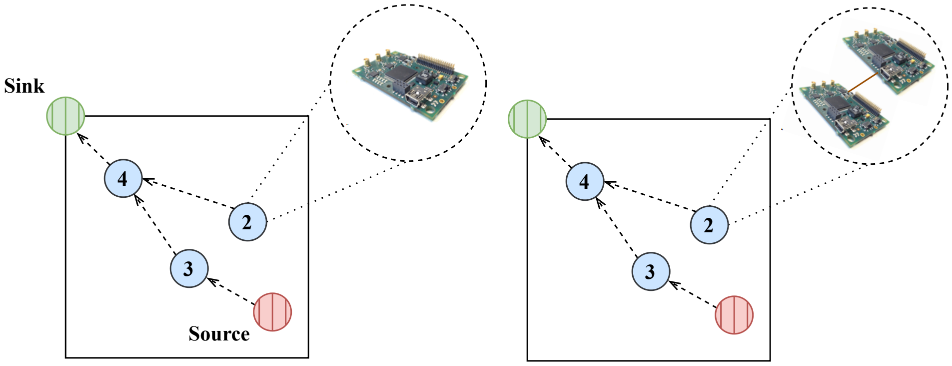


Fig. 6.9.: Logical topology configured for the proof-of-concept to compare the single-radio architecture (left graph) and our framework (right graph).

operations (i.e., transmission and receiving), and external modules (e.g., LED); plus, the power consumed by the WuRx hardware. (2) *Event Reporting Latency* as the average latency from the moment the source node detects an event and generates an event reporting packet(first event packet) to the moment the first event packet is received at the sink, and (3) *Packet Delivery Ratio (PDR)* as the ratio of event (E) and data (D) packets successfully received by the sink to the total packets transmitted by the nodes within the M2WSNs (without re-transmissions).

Finally, we have conducted three independent trials of approximately 60 minutes each, generating, on average, 896 packets per trial, to compute the metrics. For each trial, we compute the corresponding mean of the overall data collected. Then, we determine the mean of all trials. The vertical bars indicate the maximum and minimum achieved in the trials.

6.3.2 Proof-of-concept M2WSNs Implementation

We deploy a proof-of-concept testbed in a laboratory-based environment, as shown in Figure 6.8, using ContikiOS and real motes, FeuerWhere motes (FW-nodes) [69], within a well-defined topology, as shown in Figure 6.9. The testbed setup in Figure 6.8 presents the physical topology of the implemented network. This network includes five nodes, one source node (red dot) at 3-distance hop from the sink (green dot) passing via two intermediate nodes (3 and 4, blue dots), and 1-distance hop of node 2. All nodes are separated from each other, a distance between 2 to 4 meters. Figure 6.9 provides the logical topology of the network implemented, on both architectures, for packet dissemination toward the sink via a multi-hop communication. In the case of our framework, we follow the multi-hop communication process described in Figure 6.5–based on a broadcast-based wake-up scheme.

Every node is equipped with a side-by-side FW-nodes (Figure 6.9, right graph)–one FW-node when a single-radio architecture is used. The side-by-side FW-nodes is a hardware prototype consisting of one main mote and one WuRx connected by a hardware-software interface (i.e., GPIO, and the corresponding interruption handler routine implemented in the main mote). The WuRx proposed is another FW-node, but with the functionalities of a real WuRx prototype, as proposed in the literature (e.g., signal receiving tuned to 2.4GHz, triggering routines)[17], giving the lack of commercial WuRx hardware available on the market with the technical specifications of our implementation (e.g., in-band operation at 2.4Ghz) at the time the experiments were performed. However, when the WuRx hardware is ready, it can be easily connected to an FW-node, giving that the hardware-software interface is already implemented in ContikiOS. Finally, the power consumption value of the emulated WuRx is scaled to the value reported in the state-of-the-art, knowing a priori, that it will be much higher–between 3 to 6 orders of magnitude, e.g., $157mW$ to $2.2\mu W$ (-55dBm, 50m)[116])–, to the power consumed by a WuRx prototype reported in the literature (e.g., nano-watt[68]). Each FW-node is powered with a 3.7V@1300mAh battery. The sink node is connected to a PC for data logging.

We have measured the power consumption by using a Wireless Debugging and Power Measurement System–Wisdom platform–, proposed in [113] for the M2-DaGNoS validation and a TI INA219 DC Current Shunt and Power Monitor for the framework validation, using the set-up shows in Figure 6.10. The power consumption measurement of the main node and the WuR module was carried out independently to know their individual contribution throughout the system. Hence, the power consumed by the emulated WuRx can be scaled.

The event reporting latency is measured by capturing the difference of two trigger signals generated at the GPIO level by the source node (from the moment the event packet is generated) and the sink node (from the time the event packet is processed), respectively, using a digital oscilloscope, as shown in Figure 6.11a. For instances, in Figure 6.11b, the blue signal corresponds to the source node, and the red signal, to the sink node–the graph is illustrative, does not represent a measure taken from the experiments. The rise time of both pulse signals was considered in the measurement.

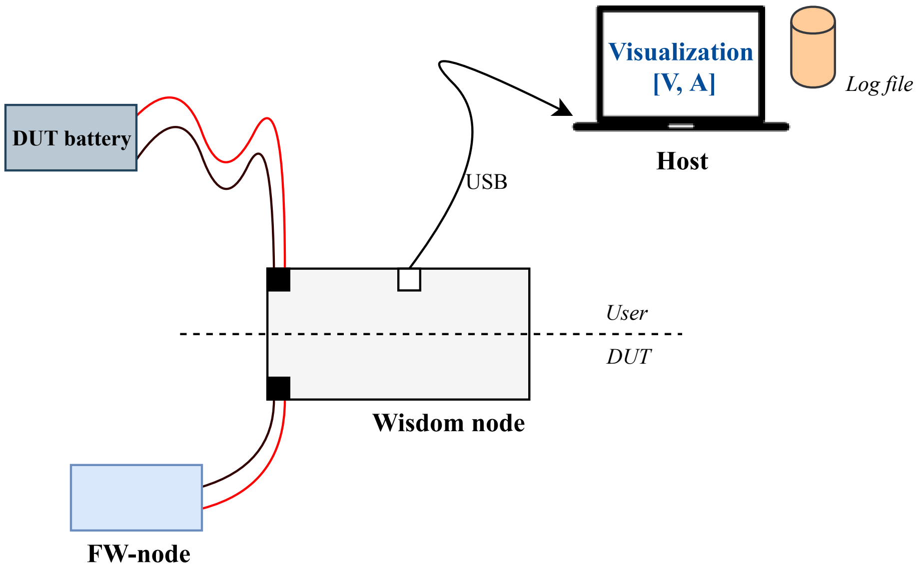
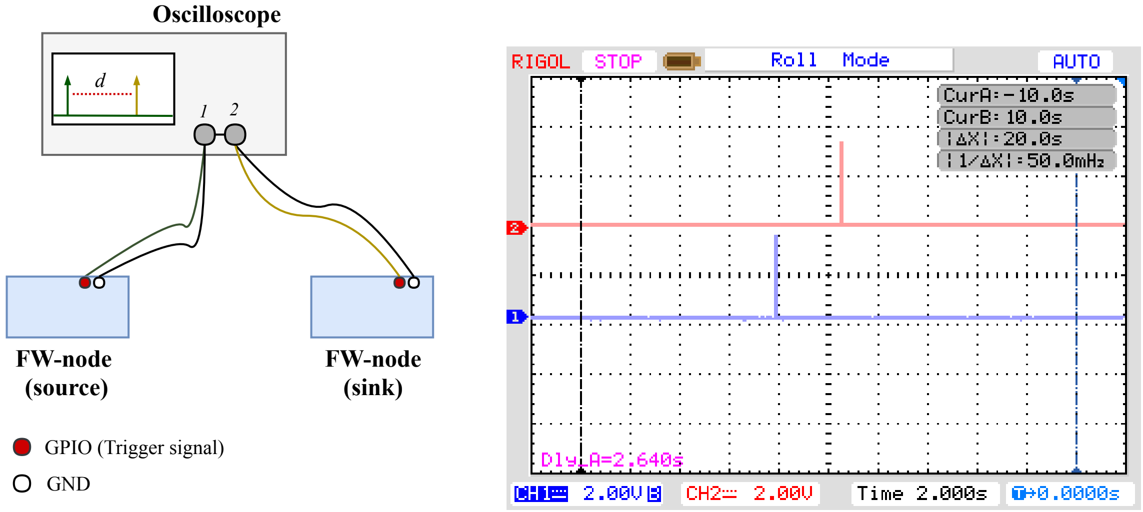


Fig. 6.10.: Power consumption measurement set-up using the Wisdom platform and the TI INA219 current sensor.



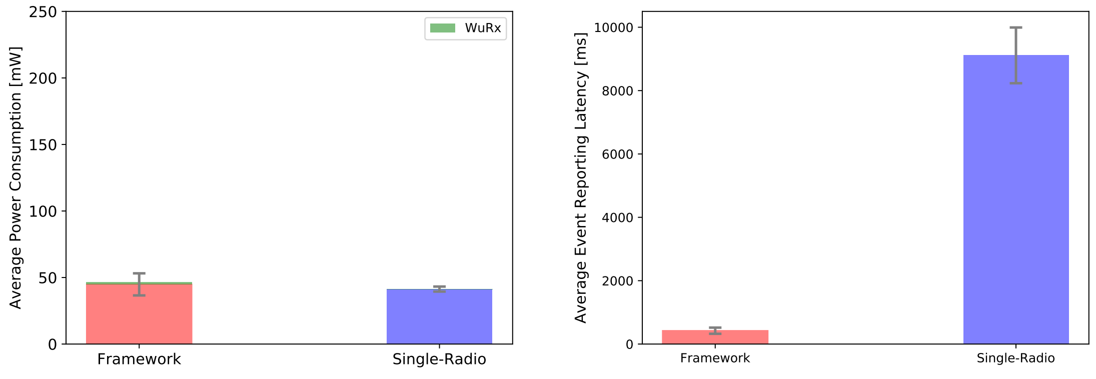
(a) Set-up diagram for latency measuring.

(b) Trigger signals. Source (blue signal), and Sink (red signal).

Fig. 6.11.: Event reporting latency measurement using trigger signals and an oscilloscope.

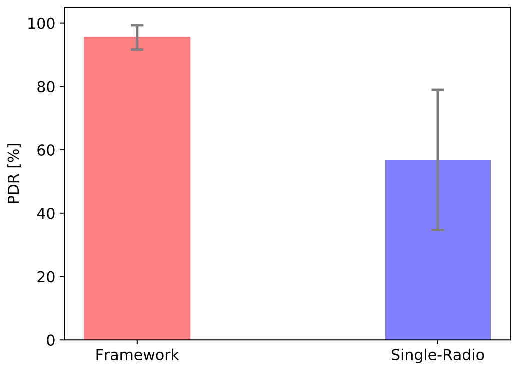
6.3.3 Validation Results

Figure 6.12a shows the average power consumption of both architectures. The results indicate that our framework consumes slightly more power ($48.31mW$) when compared to the performance of a single-radio architecture ($41.74mW$), but its performance increases in the other metrics evaluated. Though the power consumption is affected, in cases where this type of framework is required, the power consumption is comparable with a single-radio architecture based on an LDC scheme.



(a) Power Consumption.

(b) Event Reporting Latency.



(c) Packet Delivery Ratio.

Fig. 6.12.: Performance evaluation comparison between a single-radio architecture using a low-duty-cycling scheme and our framework based on a wake-radio approach.

The single-radio architecture provides a better performance regarding power consumption, due to its wake-up optimization and fast sleep methods used in its radio duty cycling implementation for packets transmission and reception, that allows to maintain the radio off approximately 99% of the time [50], but at the cost of higher latency in multi-hop communications (see Figure 6.12b). Many factors contribute to power consumption in our proposal. The 2R-MAC wake-up protocol was designed based on the “worst scenario” for wake-up and data dissemination toward the sink, i.e.,: (i) non-addressing support is used (i.e., all SNs within the neighborhood are woken up to receive a packet); (ii) every node in the path should send at least two data packets, one for wake-up the next-hop and the other for data transmission; and finally, (iii) every next hop in the known path should acknowledge the data packet. Consider, for example, the item (i). Every packet transmitted over 2.4GHz makes the main component to wake-up, i.e., its main radio is turned on to receive a packet. Hence, when the packet is not destined for the node, the process of continuous switching on the radio contributes to the total power consumption of the system (e.g., abrupt changes in the current consumption–passing from $8mA$ to $31.8mA$ to $8mA$ in microseconds). During the trials, it has been perceived that while the nodes are close, the non-designated nodes are woken-up every time a packet is transmitted over the air–effect of the broadcast-based wake-up scheme–, compared to the case when they are sufficiently separated from each other.

There are some proposals in the literature, regarding wake-up and data dissemination procedures, that provides a different approach where first all SNs within a path is woken up in an asynchronous way, then, the data packet is disseminated toward the sink following the already woken up multi-hop path as proposed in [114], or the data packet is transmitted to the sink within one-hop [83] or in few hops compare to the number of hops in the woken-up path [22]. Besides, the data packet [114] or WuS [83] might not be acknowledged. Finally, these approaches considered addressing support in their WuRx implementation, designed for event-time schemes. As future work, by considering some of these approaches, we might expect to get further improvements regarding energy-savings and latency.

We have measured the end to end delay (latency) of the first event packet generated by the source at three hops of the sink in both schemes –from source to the sink via the intermediate nodes 3 and 4 –respresent three hops to the sink. The results, in Figure 6.12b, indicate that our framework performs better than the single-radio with ContikiMAC regarding the event reporting latency. The improvement is, on average, twenty one times using a WuRx-based scheme compare to a low-duty cycling wake-up protocol, as shown in Figure 6.12b–$436.18ms$ for our framework, and 9.11s for the single-radio architecture. This difference can be due to many causes.

Under an LDC scheme, one node checks the channel for radio activities every 125ms, giving the channel check rate configured in ContikiMAC (8Hz, default value). Consequently, the source node should wait up to $125ms$ to forward a packet or to stop emitting the full data packet until the designated receiver mote wakes up and send an acknowledgment under ideal conditions. Under the effect of a real channel (over the packet transmission, e.g., collisions, back-off time, distance losses, multipath propagation), the waiting time is increased, e.g., 136.3ms for 1-hop. In a multihop communication, the latency is even bigger, giving the contribution that represents one more hop to sink and even more, the packet loss generated when it is sent toward the sink over a real channel [83]. Therefore, a mote with ContikiMAC wakes up every $125ms$ to check for radioactivity, compare to a mote with WuRx and the 2R-MAC protocol that wakes up on demand and contributes to reduce the delay due to the waiting time and transmits the data packet towards the sink faster than a LDC approach.

Another cause, regarding the difference in latency between both approaches, might be related to the number of packets introduced after switching between CMnt to EDR. The node passed to transmit every 5secs to 2.5secs (one more packet). Besides, the assisting nodes introduce packets into the network during the occurrence of an event. Hence, more packets, compared to CMnt, increases the probability of a packet buffer overload and a congestion situation into the multi-hop network, affecting the latency performance due to packet loss [117]. The latency values obtained for the single-radio architecture based on an LDC scheme are within the orders of magnitude reported in the literature when similarly configured are used, such as data-reporting frequency and number of hops [62, 118].

Figure 6.12c shows the PDR of both architectures for successful packets received by the sink during the whole operation of the M2WSNs. Our framework presents a reliability, on average, higher than 95.4%, performing better than the single-radio architecture with an LDC approach that gave a poor PDR of 49.7%. The packet loss is due to the difficulties presented during the forwarding process, where the packets are dropped or delayed due to collisions, a congestion situation in the network, effects of the communication channel (path loss and multipath propagation). Consequently, the wake-up period of the next-hop expires. Hence, the next-hop might not receive the re-transmitted packet during the wake-up period, increasing the number of re-transmissions at the source or relaying node that at the end achieves its maximum number of re-transmissions affecting the PDR. To solve this problem, the PDR can be improved by increasing the number of re-transmissions[1], but at the cost of a higher latency and energy consumption [83], due to the arise of extra congestion and collisions issues.

6.4 Remarks

In summary, this chapter provides the performance evaluation of a reliable and energy-efficiency framework for M2WSNs based on the WuR paradigm, suitable for monitoring oriented applications with low bandwidth requirements that operate simultaneously under normal circumstances and emergencies. The framework follows a layered approach, where each layer aims to fulfill specific tasks based on its information, the functions provided by its adjacent layers, and the information resulted from the cross-layer interactions. We have addressed several experiments using real motes in a laboratory-based environment to validate the performance of the framework and to compare it against a single-radio architecture based on a low-duty-cycling technique. The framework has shown better reliability in terms of the event reporting accuracy and packet-delivery ratio and significant energy savings when considering a broadcast-based wake-up scheme with one-by-one hop data transmission and acknowledgment procedures.

---

[1] The maximum CSMA retransmissions for the single-radio architecture have been configured using the default value in ContikiOS (see Table 6.2). Hence, this architecture retransmits twice compared to our framework that retransmits once–from the link-layer perspective–, contributing to reduce the power consumption without compromising the reliability of the network (PDR). The network stops working in many trials when configuring one retransmission using the single-radio architecture.

Some relevant conclusions are reached after the performance evaluation study:

- The energy consumption in both approaches might be slightly higher for some monitoring applications with ultra-power savings requirements. We have conceived in our design that all assisting nodes can enter into a deep sleeping state and exploit this capacity to further extend the energy savings in the network, i.e., with and without wake-up radios. However, the current implementation of ContikiOS ported to the FW-node does not support the deepest low power mode, i.e., LPM4, $0.1\mu A$ for MSP430 microcontrollers –currently, the microcontroller of all assisting nodes remains in its active mode during the sleeping state, representing an average current consumption of 8mA. Hence, putting the micro-controller into its deepest mode might help to improve the energy-savings. Besides, the power consumption might also be reduced by considering a WuRx scheme with addressing-support. This feature requires to add additional hardware for addressing coding and false-positive wake-up detection methods (e.g., [114]). As future work, the implementation and evaluation of these capabilities are opened to be analyzed regarding energy-efficiency and latency performance.
- The configuration considered in section 5.4 and section 6.3 are different, given that the goal in both cases has been conceived differently. In the former, the main goal has been to validate the performance of M2-DaGNoS functionalities under real motes and radio channel, employing the same experiments configuration of simulations–e.g., setting parameters, temperature profile–, considering that most of the related work remained in a simulated environment. For the latter, we have decided to evaluate and validate the performance of our framework against the typical single-radio architecture employed in most of the related work, using a "more realistic" temperature profile under the same scenario 2 (Figure 5.15) that consider the "multimodality feature" regarding data-reporting. Some minor modifications have been done in the setting parameters (Table 5.3), e.g,. $T_{gathering}$ from $50secs$ to $120secs$, reporting frequency (1 packet/10seconds to 1 packet/5seconds), and experiment time from 35mins to 60min–more workload into the network. Besides, the size of the network has been reduced (from 11 to 5 motes), due to the lack of motes at the time the experiments were performed. Therefore, we consider that the results of the trails performed, in both cases, are difficult to compare. The case is different in Section 6.3, where the trails have been performed under the same configurations for both architectures under evaluation.

Finally, to the best of our knowledge, the results obtained through this dissertation constitute the first experimental validation tests of a framework that integrates, via a cross-layer interaction, a multimodal mechanism and a network architecture based on a WuRx scheme. As future work, we expect to develop a practical evaluation platform that allows us to easily deploy applications, with a higher number of motes and tools for simultaneous measurement of power consumption and latency, and to replicate the results in a controlled manner and make strong conclusions. Finally, we consider that the main weaknesses found in previous works (Chapter 2) have been attended. Such as the implementation and evaluation of a multimodal switching mechanism in hardware and its integration into a unified asynchronous and cross-layer networking architecture based on an always-on WuRx.

Conclusions

7

Wireless Sensor Networks (WSNs) are a widely used solution for monitoring oriented applications. In general, these applications have in common the following features: Events usually occur sporadically and asynchronously, presenting rapid dynamics that in few seconds/minutes might spread or expand several km. Therefore, we need some early warning system and event tracking mechanisms along the area to react immediately upon the occurrence of an event and track it, so that, reduce its impact. Besides, these applications are usually characterized by the necessity of two data-reporting modes.

First, an event-driven (reactive) reporting (EDR) is required for event detection and tracking, usually under emergencies (e.g., a fire event). In this mode, sensor nodes can immediately react upon the event occurrence, i.e., they transmit data only when an event occurs that leads to better energy-efficient operation, but it produces low precision estimates of monitored variables over time, given its aperiodic reporting strategy.

Second, a time-driven (proactive) reporting or Continuous Monitoring (CMnt) for supervising an area, usually under normal situations (e.g., a water quality monitoring process), where sensor nodes continually monitor and report their sensed data to a sink node in a periodic fashion over time. Usually, this mode is configured with a low sampling rate, e.g., some minutes or hours, given that the variables of interest usually change slowly (e.g., temperature, relative humidity). However, having a low sampling rate implies that some events might not be detected or recognized. For instances, an emergency such as a chemical contaminant pouring situation, the CMnt scheme requires to be configured with a higher reporting frequency (e.g., seconds) that implies a high reporting accuracy and much higher estimates of the physical variable over time, allowing to detect the event, but at the cost of higher energy consumption. Hence, there is a trade-off to be made between event reporting latency and energy savings regarding these data reporting modes.

MultiModal Wireless Sensor Networks arise as a solution for the trade-off between energy savings and event reporting latency in those monitoring oriented applications where regular and emergency reporting are required simultaneously, such as forest fires and pollution monitoring. Like traditional WSNs, M2WSNs include several small and battery-powered sensor nodes (SNs), randomly deployed in an area of interest to monitor physical variables (e.g., temperature) collaboratively. Furthermore, the *multimodality* in M2WSNs allows SNs to perform data reporting in two schemes, switching between CMnt and EDR modes according to the circumstances, i.e., normal (regular) or emergency. Those mechanisms that allow SNs to switch between EDR and CMnt modes or vice versa, combining the advantages of both modes, we called as *MultiModal Switching Mechanisms*.

Most of the multimodal switching techniques help to enhance the energy-efficiency of WSNs through a switching mechanism between EDR and CMnt, assuming an always-on operation. Indeed, we got a better performance compared to applications that use only one data-reporting mode, but these techniques still present significant energy consumption and reliability issues, that can be unsuitable for applications with ultra-low-power requirements. Therefore, one approach is to combine multimodal switching mechanisms with sophisticate power-aware wake-up techniques such as low-duty cycling protocols under a single radio architecture. The low-duty-cycling techniques achieve good results regarding energy savings but making a sacrifice in the event reporting latency upon the occurrence of an event. The main reason is that the transceiver of the SNs remains most of the time off, and given that during the active operation, there are only short periods when the nodes are listening to the channel. Consequently, the sender node must wait until the receiver wakes up to transmit its data packet. Hence, a higher data latency is generated, due to the waiting time issue.

To further improve the performance of M2WSNs, i.e., better energy savings, reducing the power consumed during idle-listening and overhearing, issues presented in an low-duty-cycling approach– and at the same time to overcome the higher data latency due to the "waiting time" and collision issues, the implementation of a *Wake-up Radio* is considered. This radio has the capability to continuously monitor the wireless channel, allowing to reduce the latency while consuming a small amount of energy compared to those radios commonly employed in WSNs [94]. However, it is not enough to implement a dual-radio architecture at the physical layer, i.e., a WuR receiver and the main radio (for data transmission and reception), the network and MAC layers should be modified to support the advantages of the WuR paradigm.

7.1 Contributions

To address the aforementioned challenges, we made the following contributions in this dissertation.

MultiModal Wireless Sensor Networks. We introduced the concept of M2WSNs, from the perspective of data-reporting modes, for monitoring oriented applications that operate simultaneously under normal circumstances and emergencies. The SNs within the M2WSNs can perform data reporting in two possible schemes, CMnt for regular circumstances, and EDR for emergencies. Besides, the SNs in M2WSNs are endowed with a dual-radio architecture, network and MAC techniques based on the Wake-up-radio paradigm, promising better reliability (reporting latency) and energy savings results compared to traditional schemes.

The following publication has resulted from this contribution.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry. **MultiModal Wireless Sensor Networks for Monitoring Applications: A review**. *Journal of Circuits, Systems, and Computers*. World Scientific. Volume 29, Issue 2. 2020. pp: 1-32. ISSN:0218-1266. DOI:10.1142/S0218126620300032.

An Energy Consumption Model based on Wake-up Schemes. We introduced an energy consumption estimation model that considers the behavior and performance of wake-up protocols based on Wake-up Radio Receivers and the traditional low-duty-cycling schemes employed in M2WSNs. This model allowed us to get more insights into the differences between both approaches and for considering a multimodality feature in WSNs by combining different transceivers and reporting protocols from an energy-efficiency perspective. The following publications have resulted from this contribution.

- ARANDA, Juan; SCHOLZEL, Mario; MENDEZ, Diego; CARRILLO, Henry. **An Energy Consumption Model for MultiModal Wireless Sensor Networks based on Wake-up Radio Receivers**. In *IEEE Colombian Conference on Communications and Computing (COLCOM2018)*. Medellin, Colombia. May 16-18, 2018. IEEE Xplore Digital Library. ISBN: 978-1-5386-6820-7. DOI: 10.1109/ColComCon.2018.8466728.
- ARANDA, Juan; SCHOELZEL, Mario; MENDEZ, Diego; CARRILLO, Henry. **Multi-Modal Wireless Sensor Networks based on Wake-up Radio Receivers: An analytical Model for Energy Consumption**. *redin Revista Facultad de Ingeniería – Universidad de Antioquia*. redin. No. 91. May 2019. ISSN:0120-6230. DOI:10.17533/10.17533/udea.redin.20190401.

A MultiModal Switching Mechanism for Data-Gathering Schemes. We proposed an adaptive, distributed and asynchronous switching mechanism, M2-DaGNoS, an enhanced MultiModal Switching mechanism for Data Gathering and Node Scheduling, suitable for M2WSNs. This technique was implemented in ContikiOS, an open-source operating system for the Internet of Things, and evaluated using the Cooja emulator through an extensive simulation study and validated under real motes in a laboratory-based environment. M2-DaGNoS has presented a significant performance regarding energy savings, reporting latency, and reliability compared to state-of-the-art mechanisms. The following publications have resulted from this contribution.

- ARANDA, Juan; CARRILLO, Henry; MENDEZ, Diego. **Enhanced Multimodal Switching Mechanisms for Node Scheduling and Data Gathering in Wireless Sensor Networks**. In *IEEE Colombian Conference on Communications and Computing (COL-*

COM2017). Cartagena, Colombia. August 16-18, 2017. IEEE Xplore Digital Library. ISBN: 978-1-5386-1060-2. DOI: 10.1109/ColComCon.2017.8088194.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry; SCHOELZEL, Mario. **M2-DaGNoS: a Data Gathering and Node Scheduling Mechanism for MultiModal Wireless Sensor Networks**. *Computer Communications (ComCom) Journal*. Elsevier. Vol 148. pp. 165–175. December 2019. ISSN:0140-3664. DOI:10.1016/j.comcom.2019.09.012.

A framework for M2WSNs. We provided a reliable and energy-efficiency framework for M2WSNs based on the WuR paradigm, suitable for monitoring oriented applications with low bandwidth requirements that operate simultaneously under normal circumstances and emergencies. The framework follows a layered approach, where each layer aims to fulfill specific tasks based on its information, the functions provided by its adjacent layers, and the information resulted from the cross-layer interactions. We have addressed several experiments using real motes in a laboratory-based environment to validate the performance of the framework and to compare it against a single-radio architecture based on a low-duty-cycling technique. The framework has shown better reliability in terms of the event reporting latency and packet-delivery ratio and significant energy savings when considering the "worst-case", i.e., a broadcast-based wake-up scheme with one-by-one hop data transmission and acknowledgment procedures. The following paper has resulted from this contribution.

- ARANDA, Juan; MENDEZ, Diego; CARRILLO, Henry; SCHOELZEL, Mario. **A Framework for MultiModal Wireless Sensor Networks**. Accepted for publication in *Ad Hoc Networks Journal*. Elsevier. 2020. ISSN: 1570-8705. DOI:-.

7.2 Future Directions

We consider that the contributions of this dissertation represent an essential step towards the construction of reliable and energy-efficient M2WSNs for those monitoring oriented applications that operate simultaneously under normal circumstances and emergencies. We envisage that the proposed framework can be further improved and extended. Below, some possible directions for future work are summarized:

1. Regarding the cross-layer interactions, it remains to explore the possibility to implement the interaction between application to network, and application to link layers that might add additional features to the framework such as priority sending packet management during emergencies, dead nodes management and switching between different radio duty-cycling schemes, e.g., LDC and WuRx.

2. Regarding the M2-DaGNoS mechanism, it can be considered the case when a working node runs out of energy during a scheduling cycle by considering the integration of the node's residual energy (e.g., the voltage) in the node scheduling algorithm as input to decide when it is appropriate to communicate the withdrawal decision to the next node in node-queue Q or the neighborhood, to avoid or mitigate the problem of not monitoring the area due to lack of energy.
3. A hybrid switching medium access protocol based on a WuR sender-initiated approach (opposed to [94]) can be explore to provide support to those on-demand monitoring-oriented applications for M2WSNs with low power and latency restrictions, guarantying asynchronous and multi-hop communications and time network-wide synchronization for data-collection and node scheduling without congesting the M2WSNs while achieving high reliability and energy-savings.
4. An extended unified system and networking testbed based on the WuR paradigm can be implemented to perform evaluation studies and analysis by considering the effects of the real wireless channel when deploying dense and large-scale M2WSNs.

Bibliography

[1] Mohammad Abdul Azim, Zeyar Aung, Sofiane Moad, et al. “Energy-Efficient Methods for Highly Correlated Spatio-Temporal Environments in Wireless Sensor Network Communications”. In: *Wireless Sensor Network* 6.May (2014), pp. 67–92 (cit. on pp. 1, 2, 14, 15, 17, 19).

[2] D. Mendez, S. Diaz, and R. Kraemer. “Wireless technologies for pollution monitoring in large cities and rural areas”. In: *24th Telecommunications Forum (TELFOR)*. Nov. 2016, pp. 1–6 (cit. on pp. 1, 2, 77).

[3] I Leyva-Mayorga, M E Rivero-Angeles, C Carreto-Arellano, and V Pla. “QoS analysis for a nonpreemptive continuous monitoring and event-driven WSN protocol in mobile environments”. English. In: *International Journal of Distributed Sensor Networks* 2015 (2015) (cit. on pp. 2, 9, 12, 15, 19, 22, 32).

[4] Messaoud Doudou, Djamel Djenouri, and Nadjib Badache. “Survey on latency issues of asynchronous MAC protocols in delay-sensitive wireless sensor networks”. In: 15.2 (2013), pp. 528–550 (cit. on pp. 2, 4, 26, 27).

[5] F. Z. Djiroun and D. Djenouri. “MAC Protocols With Wake-Up Radio for Wireless Sensor Networks: A Review”. In: *IEEE Communications Surveys Tutorials* 19.1 (2017), pp. 587–618 (cit. on pp. 2, 12–14, 30–34, 47–49).

[6] B. Lee and K. Lim. “An Energy-Efficient Hybrid Data-Gathering Protocol Based on the Dynamic Switching of Reporting Schemes in Wireless Sensor Networks”. In: *IEEE Systems Journal* 6.3 (Sept. 2012), pp. 378–387 (cit. on pp. 2, 3, 9, 13, 15, 16, 18, 25, 33, 38, 66, 68, 72, 77, 79).

[7] N Bouabdallah, M E Rivero-Angeles, and B Sericola. “Continuous monitoring using event-driven reporting for cluster-based wireless sensor networks”. English. In: *IEEE Transactions on Vehicular Technology* 58 (2009), pp. 3460–3479 (cit. on pp. 2, 14, 15, 18).

[8] Xingming Sun, Chengju Xue, and Baowei Wang. “TDRHN : A Threshold Sensitive Dynamic Responsive Hybrid Network Protocol based on CTP”. In: 9.2 (2016), pp. 215–226 (cit. on pp. 2, 15, 20, 22).

[9] Carlos Eduardo R. Lopes, Fernando D. Linhares, Michele M. Santos, et al. “A Multi-tier, Multimodal Wireless Sensor Network for Environmental Monitoring”. In: *Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 589–598 (cit. on p. 2).

[10] V. Jelicic, M. Magno, D. Brunelli, G. Paci, and L. Benini. “Context-Adaptive Multimodal Wireless Sensor Network for Energy-Efficient Gas Monitoring”. In: *IEEE Sensors Journal* 13.1 (2013), pp. 328–338 (cit. on p. 2).

[11] Ibrahim Ammar, Irfan Awan, and Andrea Cullen. "Clustering synchronisation of wireless sensor network based on intersection schedules". English. In: *Simulation Modelling Practice and Theory* 60 (2016), pp. 69–89 (cit. on p. 2).

[12] Delaware County Conservation District. *What is a Watershed?* `https://www.delcocd.org/watersheds/`. [Online; retrieved 2017-11-25] (cit. on p. 3).

[13] Md Zakirul Alam Bhuiyan, Jie Wu, Guojun Wang, Tian Wang, and Mohammad Mehedi Hassan. "e-Sampling: Event-Sensitive Autonomous Adaptive Sensing and Low-Cost Monitoring in Networked Sensing Systems". In: *ACM Transactions on Autonomous and Adaptive Systems* 12.1 (2017), pp. 1–29 (cit. on pp. 4, 15, 21, 24, 66).

[15] Chirihane Gherbi, Zibouda Aliouat, and Mohamed Benmohammed. "A survey on clustering routing protocols in wireless sensor networks". In: *Sensor Review* 37.1 (2017), pp. 12–25 (cit. on pp. 12, 13).

[16] Olayinka O; Ogundile and Attahiru S Alfa. "A Survey on an Energy-Efficient and Energy-Balanced Routing Protocol forWireless Sensor Networks". In: *Sensors* 17 (2017), pp. 1–51 (cit. on pp. 12, 14).

[17] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli. "Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey". In: *IEEE Communications Surveys Tutorials* PP.99 (2017), pp. 1–1 (cit. on pp. 12, 28–34, 47–49, 62, 94, 104).

[18] R C Carrano, D Passos, L C S Magalhaes, and C V N Albuquerque. "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks". English. In: *IEEE Communications Surveys and Tutorials* 16.1 (2014), pp. 181–194 (cit. on pp. 12, 25–28, 33).

[19] Vishal Krishna Singh, Vivek Kumar Singh, and Manish Kumar. "In-Network Data Processing Based on Compressed Sensing in WSN: A Survey". In: *Wireless Personal Communications* (2017), pp. 1–38 (cit. on pp. 12, 14).

[20] M. Mehdi Afsar and Mohammad H. Tayarani-N. "Clustering in sensor networks: A literature survey". In: *Journal of Network and Computer Applications* 46 (2014), pp. 198–226 (cit. on pp. 13, 26, 91).

[21] Ayub Shokrollahi and Babak Mazloom-Nezhad Maybodi. "An Energy-Efficient Clustering Algorithm Using Fuzzy C-Means and Genetic Fuzzy System for Wireless Sensor Network". In: *Journal of Circuits, Systems and Computers* 26.01 (2017), pp. 1–22 (cit. on p. 13).

[22] 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". In: *Ad Hoc Networks* 59 (2017), pp. 97–115 (cit. on pp. 14, 31, 48–50, 94, 95, 100, 107).

[23] Juha Petäjäjärvi, Konstantin Mikhaylov, Risto Vuohtoniemi, Heikki Karvonen, and Jari Iinatti. "On the human body communications: wake-up receiver design and channel characterization". In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016), p. 179 (cit. on p. 14).

[24] A. Adulyasas, Zhili Sun, and Ning Wang. "An event-driven clustering-based technique for data monitoring in wireless sensor networks". In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pp. 653–656 (cit. on pp. 15, 16, 18).

[25] Yimei Kang, Bin Hu, Yi Ding, and Jingdong Tan. “A hybrid node scheduling approach based on energy efficient chain routing for WSN”. English. In: *Advances in Mechanical Engineering* 2014 (2014), pp. 1–12 (cit. on pp. 15, 17, 19, 23, 38, 66–68, 70–72, 74, 77–79, 102, 129, 130).

[26] Y Hu, Y Niu, J Lam, and Z Shu. *An Energy-Efficient Adaptive Overlapping Clustering Method for Dynamic Continuous Monitoring in WSNs*. 2017 (cit. on pp. 15, 20, 23).

[27] R. Nagarajan and R. Dhanasekaran. “Energy Efficient Data Transmission Approaches for Wireless Industrial Automation”. In: *Current Signal Transduction Therapy* 13.1 (2018), pp. 37–43 (cit. on pp. 15, 21, 25).

[28] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, et al. “CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks”. In: *ACM Trans. Sen. Netw.* 10.1 (Dec. 2013), 16:1–16:49 (cit. on p. 22).

[29] S. Basagni, C. Petrioli, and D. Spenza. “CTP-WUR: The collection tree protocol in wake-up radio WSNs for critical applications”. In: *International Conference on Computing, Networking and Communications (ICNC)*. Feb. 2016, pp. 1–6 (cit. on pp. 23, 34).

[30] A Adulyasas. “Connected coverage assurance for sensor scheduling in wireless sensor networks.” PhD thesis. Dept. Elect. Eng, University of Surrey, Guildford, Surrey, 2015, pp. 1–197 (cit. on p. 25).

[31] Alvin C. Valera, W.-S. Wee Seng Soh, and H.-P. Hwee Pink Tan. “Survey on wakeup scheduling for environmentally-powered wireless sensor networks”. English. In: *Computer Communications* 52 (2014), pp. 21–36 (cit. on pp. 25, 26).

[32] Amulya Ratna Swain and R. C. Hansdah. “A model for the classification and survey of clock synchronization protocols in WSNs”. English. In: *Ad Hoc Networks* 27 (2015), pp. 219–241 (cit. on pp. 26, 27).

[33] D. Djenouri and M. Bagaa. “Synchronization Protocols and Implementation Issues in Wireless Sensor Networks: A Review”. In: *IEEE Systems Journal* 10.2 (June 2016), pp. 617–627 (cit. on pp. 26, 27).

[34] Kai Han, J Luo, Y Liu, and AV Vasilakos. “Algorithm Design for Data Communications in Duty-Cycled Wireless Sensor Networks: A Survey”. In: *IEEE Communications Magazine* July (2013), pp. 107–113 (cit. on pp. 26, 28).

[35] Sergio Diaz, Diego Mendez, and Rolf Kraemer. “A Review on Self-Healing and Self-Organizing Techniques for Wireless Sensor Networks”. In: *Journal of Circuits, Systems and Computers* (2018), pp. 1–36 (cit. on p. 26).

[36] Geetika Dhand and S.S. Tyagi. “Data Aggregation Techniques in WSN: Survey”. In: *Procedia Computer Science* 92 (2016), pp. 378–384 (cit. on p. 26).

[37] Jie Hao, Baoxian Zhang, and Hussein Mouftah. “Routing Protocols for Duty Cycled Wireless Sensor Networks: A Survey”. In: *IEEE Communications Magazine* December (2012), pp. 116–123 (cit. on p. 26).

[38] Nikolaos A. Pantazis, Stefanos A. Nikolidakis, and Dimitrios D. Vergados. “Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey”. In: *IEEE Communications Surveys & Tutorials* 15.2 (2013), pp. 551–591 (cit. on p. 26).

[39]Amritpal Pandey Gaurav;Singh. “Recent Advancements in Energy Efficient Routing in Wireless Sensor Networks: A Survey”. In: *Advances in Intelligent Systems and Computing* 436 (2016), pp. 367–376 (cit. on p. 26).

[40]C Li, H Zhang, B Hao, and J Li. “A Survey on routing protocols for large-scale wireless sensor networks”. English. In: *Sensors* 11.4 (2011), pp. 3498–3526 (cit. on p. 26).

[41]Chuan Zhu, Chunlin Zheng, Lei Shu, and Guangjie Han. “A survey on coverage and connectivity issues in wireless sensor networks”. In: *Journal of Network and Computer Applications* 35.2 (2012), pp. 619–632 (cit. on p. 26).

[42]S H Lee and L Choi. “A+MAC: A streamlined variable duty-cycle MAC protocol for wireless sensor networks”. English. In: *International Journal of Distributed Sensor Networks* 2013 (2013) (cit. on pp. 26, 28).

[43]Dishee Agarwal and Arvind Kakria. “Survey of Asynchronous Medium Access Protocols for Wireless Sensor Networks”. In: *International Journal of Computer Science and Technology* 6.2 (2015), pp. 326–330 (cit. on p. 26).

[44]L Liang, X Liu, Y Wang, W Feng, and G Yang. “SW-MAC: A low-latency MAC protocol with adaptive sleeping for wireless sensor networks”. In: *Wireless Personal Communications* 77.2 (2014), pp. 1191–1211 (cit. on p. 26).

[45]Guijuan Wang, Jiguo Yu, Dongxiao Yu, et al. “DS-MAC: An energy efficient demand sleep MAC protocol with low latency for wireless sensor networks sensor networks”. In: *Journal of Network and Computer Applications* 58 (2015), pp. 309–326 (cit. on p. 27).

[46]Thanh Dinh, Younghan Kim, Tao Gu, and Athanasios V. Vasilakos. “L-MAC: A wake-up time self-learning MAC protocol for wireless sensor networks”. In: *Computer Networks* 105 (2016), pp. 33–46 (cit. on pp. 27, 28).

[47]P. Huang, C. J. Liu, and L. Xiao. “TAS-MAC: A traffic-adaptive synchronous MAC protocol for wireless sensor networks”. In: (June 2013), pp. 113–121 (cit. on pp. 27, 28).

[48]B Marques and M Ricardo. “Energy-efficient node selection in application-driven WSN”. In: *Wireless Networks* (2016), pp. 1–30 (cit. on pp. 27, 28).

[49]Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. “X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks”. In: SenSys ’06. Boulder, Colorado, USA: ACM, 2006, pp. 307–320 (cit. on pp. 27, 28).

[50]Adam Dunkels. “The ContikiMAC Radio Duty Cycling Protocol”. In: *SICS Technical Report T2011:13 , ISSN 1100-3154* (2011), pp. 1–11 (cit. on pp. 27, 28, 69, 76, 101, 106, 129).

[51]Mathieu Michel and Bruno Quoitin. “Technical Report : ContikiMAC vs X-MAC performance analysis”. In: (2015), pp. 1–28. eprint: `1404.3589` (cit. on pp. 27, 76, 101).

[52]M. F. Youssef, K. M. F. Elsayed, and A. H. Zahran. “Contiki-AMAC –The enhanced adaptive radio duty cycling protocol: Proposal and analysis”. In: *International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*. Apr. 2016, pp. 1–6 (cit. on p. 27).

[53]M Doudou, D Djenouri, J M Barcelo-Ordinas, and N Badache. “Delay-efficient MAC protocol with traffic differentiation and run-time parameter adaptation for energy-constrained wireless sensor networks”. In: *Wireless Networks* 22.2 (2016), pp. 467–490 (cit. on p. 28).

[54] Jun Bum Lim, Beakcheol Jang, and Mihail L. Sichitiu. “MCAS-MAC: A multichannel asynchronous scheduled MAC protocol for wireless sensor networks”. In: *Computer Communications* 56 (2015), pp. 98–107 (cit. on p. 28).

[55] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. “Efficient network flooding and time synchronization with Glossy”. In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks* (2011), pp. 73–84 (cit. on p. 28).

[56] J.-H. Wang, Y Yu, and M C Nnaemeka. “A high throughput MAC protocol for wireless sensor networks in surveillance applications”. In: *Journal of Networks* 8.9 (2013), pp. 2005–2012 (cit. on p. 28).

[57] Mirko R. Kosanovic and Mile K. Stojcev. “RPATS - Reliable power aware time synchronization protocol”. In: *Microelectronics Reliability* 54.1 (2014), pp. 303–315 (cit. on p. 28).

[58] W Bober and C J Bleakley. “BailighPulse: A low duty cycle data gathering protocol for mostly-off Wireless Sensor Networks”. In: *Computer Networks* 69 (2014), pp. 51–65 (cit. on p. 28).

[59] Y Chen, F Qin, and W Yi. “Guard beacon: An energy-efficient beacon strategy for time synchronization in wireless sensor networks”. In: *IEEE Communications Letters* 18.6 (2014), pp. 987–990 (cit. on p. 28).

[60] Zhichao Cao, Yuan He, and Yunhao Liu. “L^2: Lazy Forwarding in Low-Duty-Cycle”. In: *IEEE/ACM Transactions on Networking* 23.3 (2015), pp. 922–930 (cit. on p. 28).

[61] Beakcheol Jang, Jun Bum, Mihail L Sichitiu, J B Lim, and Mihail L Sichitiu. “An asynchronous scheduled MAC protocol for wireless sensor networks”. In: *Computer Networks* 57.1 (2013), pp. 85–98 (cit. on p. 28).

[62] Joaquim Oller, Ilker Demirkol, Jordi Casademont, et al. “Has Time Come to Switch from Duty-Cycled MAC Protocols to Wake-Up Radio for Wireless Sensor Networks?” English. In: *IEEE/ACM Transactions on Networking* 24.2 (2016), pp. 674–687 (cit. on pp. 28–33, 47, 48, 108).

[63] Vana Jelicic, Michele Magno, Davide Brunelli, Vedran Bilas, and Luca Benini. “Analytic comparison of wake-up receivers for WSNs and benefits over the wake-on radio scheme”. In: *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N ’12* (2012), p. 99 (cit. on p. 29).

[64] V Jelicic, M Magno, D Brunelli, V Bilas, and L Benini. “Benefits of Wake-Up Radio in Energy-Efficient Multimodal Surveillance Wireless Sensor Network”. English. In: *IEEE Sensors Journal* 14.9 (2014), pp. 3210–3220 (cit. on p. 29).

[65] Florin Hutu, Aissa Khoumeri, Guillaume Villemaud, and Jean-Marie Gorce. “A new wake-up radio architecture for wireless sensor networks”. In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014), p. 177 (cit. on p. 29).

[66] Chandra Shekhar, Shirshu Varma, and M. Radhakrishna. “A Passive Wake-Up Circuit for Event Driven Wireless Sensor Network Applications”. In: *Journal of Circuits, Systems and Computers* 24.08 (2015), pp. 1–15 (cit. on p. 29).

[67] Dora Spenza, Michele Magno, Stefano Basagni, et al. “Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems”. In: *Proceedings - IEEE INFOCOM* 26 (2015), pp. 522–530 (cit. on p. 29).

[68] A. Elgani, M. Magno, F. Renzini, et al. “Nanowatt Wake-Up Radios: Discrete-Components and Integrated Architectures”. In: *25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2018, pp. 793–796 (cit. on pp. 29, 104).

[69] K. Piotrowski, S. Ortmann, and P. Langendörfer. “Multi-radio wireless sensor node for mobile biomedical monitoring”. In: *Biomedical Engineering / Biomedizinische Technik*. Vol. 57. SI-1 Track-L. 2012, pp. 725–728 (cit. on pp. 29, 86, 87, 96, 103).

[70] Debasish Ghose, Vicent Pla, and Frank Y. Li. “MAC Protocols for Wake-Up Radio: Principles, Modeling and Performance Analysis”. In: *IEEE Transactions on Industrial Informatics* 14.5 (2018), pp. 2294–2306 (cit. on pp. 30, 31).

[71] Falko Dressler, Margit Mutschlechner, Bijun Li, et al. “Monitoring Bats in the Wild: On Using Erasure Codes for Energy-Efficient Wireless Sensor Networks”. In: *ACM Transactions on Sensor Networks* 12.1 (2016), pp. 1–29 (cit. on p. 31).

[72] Matthias Vodel, Mirko Lippmann, and Wolfram Hardt. “WRTA - Wake-Up-Receiver optimised routing and topology optimisation approach”. In: *12th International Conference on ITS Telecommunications (ITST)* (2012), pp. 329–334 (cit. on p. 31).

[73] Luca Catarinucci, Riccardo Colella, Giuseppe Del Fiore, et al. “A cross-layer approach to minimize the energy consumption in wireless sensor networks”. In: *International Journal of Distributed Sensor Networks* 2014 (2014) (cit. on p. 31).

[74] Li Chen, Jeremy Warner, Wendi Heinzelman, and Ilker Demirkol. “MH-REACH-Mote: Supporting multi-hop passive radio wake-up for wireless sensor networks”. In: *IEEE International Conference on Communications* 2015-Septe (2015), pp. 6512–6518 (cit. on pp. 31, 32).

[75] F Sutton, B Buchli, J Beutel, and L Thiele. “Zippy: On-demand network flooding”. In: *13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015* (2015), pp. 45–58 (cit. on p. 31).

[76] Mikko Valta, Pekka Koskela, and Jouni Hiltunen. “Wake-up radio implementation for internet of things”. In: *Int. J. Autonomous and Adaptive Communications Systems* 9.1/2 (2016), pp. 85–102 (cit. on p. 31).

[77] Timo Kumberg, Christian Schindelhauer, and Leonhard Reindl. “Exploiting Concurrent Wake-Up Transmissions Using Beat Frequencies”. In: *Sensors* 17.8 (2017), pp. 1–21 (cit. on p. 32).

[78] Michele Magno and Luca Benini. “An ultra low power high sensitivity wake-up radio receiver with addressing capability”. In: *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2014), pp. 92–99 (cit. on p. 32).

[79] M. Magno, S. Marinkovic, B. Srbinovski, and E. M. Popovici. “Wake-up radio receiver based power minimization techniques for wireless sensor networks: A review”. In: *Microelectronics Journal* 45.12 (2014), pp. 1627–1633 (cit. on p. 32).

[80] Xuefeng Liu, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. “Enabling Reliable and Network-Wide Wakeup in Wireless Sensor Networks”. In: *IEEE Transactions on Wireless Communications* 15.3 (2016), pp. 2262–2275 (cit. on p. 32).

[81] Masanori Monobe and Hiroyuki Yomo. “Reducing wake-up overhead for energy-efficient on-demand wireless sensor networks”. In: *IEEE Globecom Workshops– Proceedings* (2016) (cit. on p. 32).

[82] Rajeev Piyare, Timofei Istomin, and Amy L. Murphy. “WaCo: A Wake-Up Radio COOJA Extension for Simulating Ultra Low Power Radios”. In: *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. Uppsala, Sweden: Junction Publishing, pp. 48–53 (cit. on pp. 33, 42, 50, 76, 97, 98).

[83] S. L. Sampayo, J. Montavont, F. Prégaldiny, and T. Noël. “Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?” In: *14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2018, pp. 1–8 (cit. on pp. 34, 47, 62–64, 76, 81, 86, 90, 100, 101, 107, 108).

[84] Felix Sutton, Reto Da Forno, Jan Beutel, and Lothar Thiele. “BLITZ : A Network Architecture for Low Latency and Energy-efficient Event-triggered Wireless Communication”. In: *Session 6: Low-Power Wireless Networks HotWireless’17*. 2017, pp. 55–59 (cit. on p. 35).

[85] Rajeev Ranjan and Shirshu Varma. “Challenges and Implementation on Cross Layer Design for Wireless Sensor Networks”. In: *Wireless Personal Communications* 86.2 (2016), pp. 1037–1060 (cit. on pp. 36, 43).

[86] “IEEE Standard for Low-Rate Wireless Networks”. In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709 (cit. on pp. 37, 41, 42).

[87] Gnawali, Omprakash and Fonseca, Rodrigo and Jamieson, Kyle and Moss, David and Levis, Philip. “Collection Tree Protocol”. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. SenSys ’09. ACM, 2009, pp. 1–14 (cit. on p. 39).

[88] A. Dunkels, B. Gronvall, and T. Voigt. “Contiki - a lightweight and flexible operating system for tiny networked sensors”. In: *29th Annual IEEE International Conference on Local Computer Networks*. Oct. 2004, pp. 455–462 (cit. on p. 39).

[89] Adam Dunkels. “Rime — A Lightweight Layered Communication Stack for Sensor Networks”. In: *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*. Delft, The Netherlands, Jan. 2007 (cit. on pp. 39, 68, 76, 129).

[90] W. Van Heddeghem; “Cross-layer Link Estimation For Contiki-based Wireless Sensor Networks”. Master’s Thesis. Vrije Universiteit Brussel, 2009, p. 96 (cit. on pp. 39, 40, 71, 72).

[91] Hamadoun Tall, Gerard Chalhoub, and Michel Misson. “Implementation and performance evaluation of IEEE 802.15.4 unslotted CSMA/CA protocol on Contiki OS”. In: *Annals of Telecommunications* 71.9 (2016), pp. 517–526 (cit. on pp. 41, 42).

[92] R. Piyare, A. L. Murphy, M. Magno, and L. Benini. “KRATOS: An Open Source Hardware-Software Platform for Rapid Research in LPWANs”. In: *14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2018, pp. 1–4 (cit. on pp. 42, 94, 98).

[93] Kévin Roussel and Ye-Qiong Song. “A critical analysis of Contiki’s network stack for integrating new MAC protocols”. In: RR-8776 (2013), p. 13 (cit. on p. 42).

[94] Rajeev Piyare, Amy L. Murphy, Michele Magno, and Luca Benini. “On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT”. In: *Sensors* 18.11 (2018) (cit. on pp. 43, 93–95, 112, 115).

[95] Nafiseh Seyed Mazloum and Ove Edfors. “Influence of Duty-Cycled Wake-Up Receiver Characteristics on Energy Consumption in Single-Hop Networks”. In: *IEEE Transactions on Wireless Communications* 16.6 (2017), pp. 3870–3884 (cit. on pp. 49, 52, 62, 64).

[96] Maarten Lont, Dusan Milosevic, Peter G M Baltus, Arthur H M Van Roermund, and Guido Dolmans. “Analytical models for the wake-up receiver power budget for wireless sensor networks”. In: *GLOBECOM - IEEE Global Telecommunications Conference* (2009) (cit. on pp. 49, 52, 54).

[97] Yan Zhang, Li Huang, Guido Dolmans, and Harmke de Groot. “An analytical model for energy efficiency analysis of different wakeup radio schemes”. In: *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications* (2009), pp. 1148–1152 (cit. on p. 49).

[98] M. Zhang, D. Ghose, and F. Y. Li. “Does Wake-Up Radio Always Consume Lower Energy Than Duty-Cycled Protocols?” In: *IEEE 86th Vehicular Technology Conference (VTC-Fall)*. Sept. 2017, pp. 1–5 (cit. on p. 49).

[99] *MSP430F543xA, MSP430F541xA Mixed-Signal Microcontrollers*. SLAS655F. Rev. 3. Texas Instruments. July 2018 (cit. on pp. 52, 59, 64).

[100] David Santos, Max Frohberg, and Mario Schölzel. “Integration of IHP-node and Wakeup-Receiver”. In: CoGeWSN Workshop Nuremberg-Erlangen, Germany, 2017 (cit. on pp. 52, 59, 64).

[101] Moshaddique Al Ameen, Niamat Ullah, M Sanaullah Chowdhury, SM Riazul Islam, and Kyung-sup Kwak. “A power efficient MAC protocol for wireless body area networks”. In: *EURASIP Journal on Wireless Communications and Networking* 2012.1 (2012), p. 33 (cit. on p. 52).

[102] Marie-Paule Uwase, Maite Bezunartea, Jacques Tiberghien, Jean-Michel Dricot, and Kris Steenhaut. “Experimental Comparison of Radio Duty Cycling Protocols for Wireless Sensor Networks”. In: *IEEE Sensors Journal* 17.19 (2017), pp. 6474–6482 (cit. on p. 56).

[103] *CC1101-Low-Power Sub-1 GHz RF Transceiver*. SWRS061I. Texas Instruments. 2018 (cit. on p. 59).

[104] M. Magno, V. Jelicic, B. Srbinovski, et al. “Design, Implementation, and Performance Evaluation of a Flexible Low-Latency Nanowatt Wake-Up Radio Receiver”. In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 633–644 (cit. on p. 62).

[105] S. Diaz, D. Mendez, and M. Schölzel. “Dynamic Gallager-Humblet-Spira Algorithm for Wireless Sensor Networks”. In: *IEEE Colombian Conference on Communications and Computing (COLCOM)*. 2018, pp. 1–6 (cit. on pp. 68, 71, 91).

[106] Contiki. *Contiki: The Open Source OS for the Internet of Things*. `http://www.contiki-os.org/`. [Online; retrieved 2019-04-02] (cit. on p. 75).

[107] Shujuan Chen, Adam Dunkels, Fredrik Osterlind, Thiemo Voigt, and Mikael Johansson. “Time synchronization for predictable and secure data collection in wireless sensor networks”. In: *In The Sixth Annual Mediterranean Ad Hoc Networking WorkShop*. 2007, pp. 165–172 (cit. on p. 76).

[108] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. “Powertrace: Network-level Power Profiling for Low-power Wireless Networks Low-power Wireless”. In: *SICS Technical Report T2011:05* (2011), p. 14 (cit. on p. 81).

[109] K. Mahendrababu and K. L. Joshitha. “A solution to energy hole problem in Wireless Sensor Networks using WITRICITY”. In: *International Conference on Information Communication and Embedded Systems*. 2014, pp. 1–6 (cit. on p. 86).

[110] P. Chatterjee and N. Das. “Multiple sink deployment in multi-hop wireless sensor networks to enhance lifetime”. In: *Applications and Innovations in Mobile Computing (AIMoC)*. 2015, pp. 48–54 (cit. on p. 86).

[111] B. Yahya and J. Ben-Othman. “REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks”. In: *IEEE Global Telecommunications Conference*. 2009, pp. 1–7 (cit. on p. 86).

[112] M. Meng, X. Wu, H. Xu, et al. “Energy Efficient Routing in Multiple Sink Sensor Networks”. In: *International Conference on Computational Science and its Applications*. 2007, pp. 561–566 (cit. on p. 86).

[113] M. Frohberg, E. Batlkhagva, and M. Schölzel. “WISDOM — A wireless debugging and power measurement system for field tests and device observation in WSN”. In: *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. 2017, pp. 319–324 (cit. on pp. 88, 104).

[114] Felix Sutton, Reto Da Forno, Jan Beutel, and Lothar Thiele. “BLITZ: Low Latency and Energy-Efficient Communication for Event-Triggered Wireless Sensing Systems”. In: *ACM Trans. Sen. Netw.* 15.2 (Mar. 2019), 25:1–25:38 (cit. on pp. 94, 95, 100, 107, 109).

[115] Kévin Roussel, Ye-Qiong Song, and Olivier Zendra. “Lessons Learned through Implementation and Performance Comparison of Two MAC/RDC Protocols on Different WSN OS”. In: *INRIA Nancy.* RR-8777.hal-01202664 (2015), pp. 1–25 (cit. on p. 101).

[116] M. Magno, V. Jelicic, B. Srbinovski, et al. “Design, Implementation, and Performance Evaluation of a Flexible Low-Latency Nanowatt Wake-Up Radio Receiver”. In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 633–644 (cit. on p. 104).

[117] Omer Chughtai, Nasreen Badruddin, Maaz Rehan, and Abid Khan. “Congestion detection and alleviation in multihop wireless sensor networks”. In: *Wireless Communications and Mobile Computing* 2017 (2017), pp. 1–13 (cit. on p. 108).

[118] Maite Bezunartea, Mai Banh, Miguel Gamallo, Jacques Tiberghien, and Kris Steenhaut. “Impact of Cross-Layer Interactions between Radio Duty Cycling and Routing on the Efficiency of a Wireless Sensor Network: A Testbed Study Involving ContikiMAC and RPL”. In: *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*. ICC ’17. Cambridge, United Kingdom: Association for Computing Machinery, 2017 (cit. on p. 108).

Colophon

This thesis was typeset with $\LaTeX\,2_\varepsilon$. It is based on the *Clean Thesis* style developed by Ricardo Langner.

Appendices

A

Enhanced Hybrid Node Scheduling Mechanism Implementation

In this appendix, we present the implementation of an **e**nhanced **H**ybrid **N**ode **S**cheduling mechanism, eHNS, which is a better version of the HNS mechanism proposed in [25] and described in Chapter 2, regarding energy-efficiency. The eHNS is implemented in ContikiOS, and its performance evaluation against HNS and M2-DaGNoS mechanisms has been done in Chapter 5.

Most of the multimodal switching mechanisms proposed in the literature (see Table 2.3) aim to improve the energy-efficiency of WSNs by switching between EDR and CMnt, assuming an always-on operation. Indeed, we got a better performance compared to applications that only use one data-reporting mode, but it still presents significant energy consumption that can be unsuitable for low-power applications. Therefore, we implemented eHNS with a radio duty-cycling mechanism –i.e., ContikiMAC–, to further improve the switching algorithms in terms of energy consumption. The eHNS was modeled based on the algorithms presented in [25], using the state machine model shown in Figure A.1. The algorithms were implemented from scratch in ContikiOS, given that the authors did not release the codes.

From an application perspective, the eHNS mechanism goes through different states. All sensor nodes start at the *Sleeping* state, and after settling the network, SNs go to the *Initializing* state, where they exchange broadcast messages to decide which node is going to be the first *working node* (WN) in a grid and be in charge of building a node queue. The other nodes remain as *redundant nodes* until it is their turn to be a WN or become an *assistant node* (i.e. switch to *Assisting* state) when an event enters a grid. The working role is rotated between nodes inside a grid, according to a cyclical node sleep scheduling technique, and switch to the *assisting* state according to an event-driven mode wake-up technique proposed in [25] and described in Chapter 2.

As mentioned before, we developed eHNS mechanism under the ContikiOS. Hence, we employed the RIME stack [89] under two radio-duty cycling (RDC) approaches: always-on and duty-cycling. For the latter, we used an asynchronous duty-cycling technique, ContikiMAC [50], also available in ContikiOS. The RIME offers us robust ready-to-use modules such as broadcast, unicast, multihop, route and neighbor discovery modules– modules widely chosen in real WSNs applications. Figure A.2 presents the protocol stacks implemented for eHNS. The boxes highlighted in gray are drivers (techniques, available in ContikiOS) not considered by the authors in [25], which are important for real implementations. By adopting these modules, we solved most of the issues presented in [25].

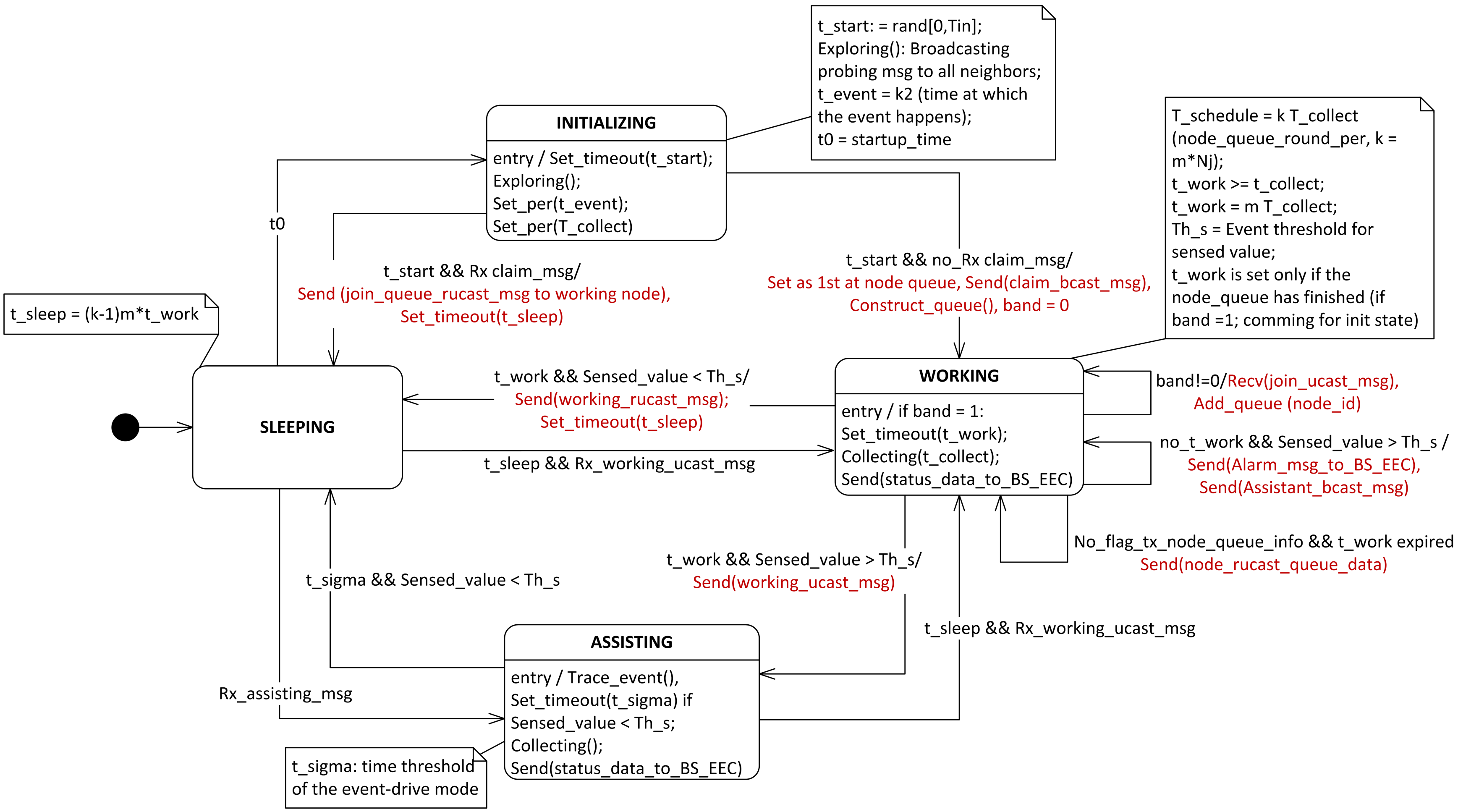


Fig. A.1.: eHNS state machine model (Adapted from [25]).

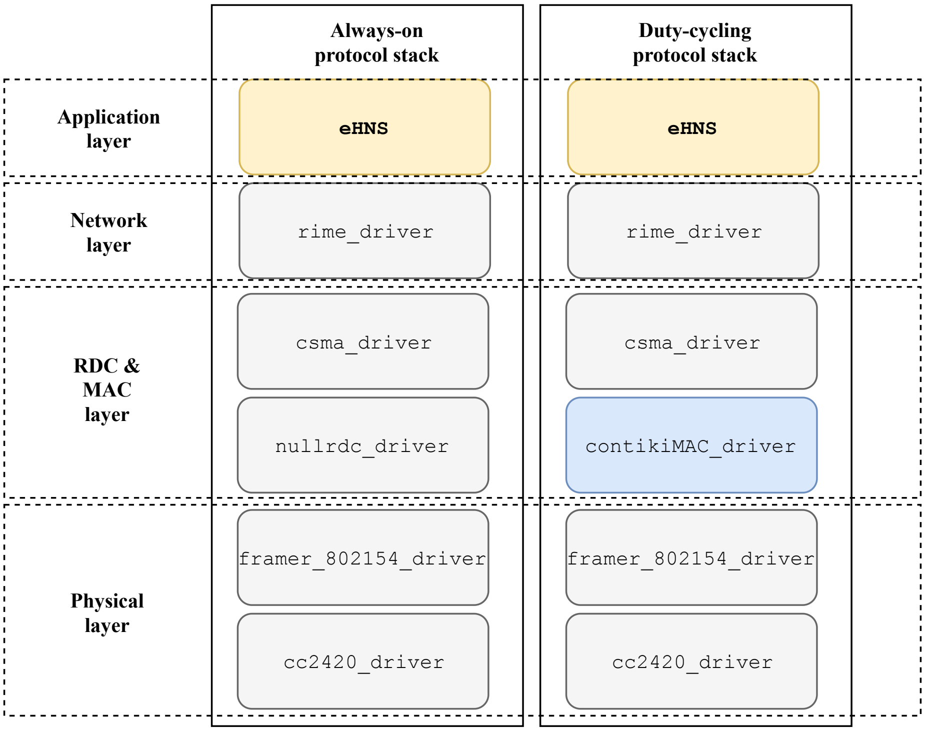


Fig. A.2.: Protocol stacks: Always-on and Duty-cycling modes.