Ad Hoc Networks 106 (2020) 102201

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A framework for multimodal wireless sensor networks

Juan Aranda[a,b,*], Diego Mendez[b], Henry Carrillo[a], Mario Schölzel[c]

[a] Exact Sciences and Engineering School, Universidad Sergio Arboleda, Calle 74 No. 14–14, 110221 Bogotá, Colombia
[b] Department of Electronics Engineering, Pontificia Universidad Javeriana, Cra. 7 No. 40–62 José Gabriel Maldonado S.J. Bldg., 110231 Bogotá, Colombia
[c] Institut für Informatik, Automatisierung und Elektronik, Hochschule Nordhausen, Weinberghof 4, 99734 Nordhausen, Germany

ARTICLE INFO

Article history:
Received 15 October 2019
Revised 14 March 2020
Accepted 5 May 2020
Available online 18 May 2020

Keywords:
Cross-layer design
Energy-efficiency
Internet of things
Low-duty-cycling
Reliability
Wake-up radio
Wake-up receiver
Testbed

ABSTRACT

During the last decade, the Wireless Sensor Networks have been considered as a broad solution for Internet of Things (IoT) based systems such as noise and air pollution monitoring systems. These systems should be able to monitor the physical variables regularly and simultaneously, to react immediately upon the occurrence of an emergency, and to report the event and its associated data to an observer, in a reliable and energy-efficient way. Recently, the MultiModal Wireless Sensor Networks (M2WSNs) have been proposed as a solution for monitoring oriented applications with low bandwidth requirements that operate simultaneously under normal circumstances and emergencies. In this paper, we present a reliable and energy-efficiency framework for M2WSNs based on the IoT and the Wake-up Radio paradigms. The framework is implemented in ContikiOS, an open-source operating system for IoT solutions, and its performance is evaluated in hardware in an indoor testbed. The experimental results show that the proposed framework provides better reliability in terms of the event reporting latency and packet delivery ratio. Besides, a significant energy-saving is achieved when considering a broadcast-based wake-up scheme with one communication hop for data transmission and acknowledgment procedures in a multi-hop network, compared to a single-radio approach based on a traditional low-duty-cycling and networking techniques.

1. Introduction

MultiModal Wireless Sensor Networks (M2WSNs) are a solution proposed for those monitoring-oriented applications where simultaneously need to manage normal circumstances and emergencies via a reliable and energy-efficient multi-hop communication [1]. Like traditional wireless sensor networks (WSNs), the M2WSNs include several sensor nodes (SNs) with limited energy and computational resources, which 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 [1].

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 low-duty-cycling (LDC) protocol [2], the implementation of *Wake-up Radios (WuR)* 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 (3 to 6 orders of magnitude less [3]) compared to those radios commonly employed in WSNs. However, it is not enough to implement a dual-radio architecture at the physical layer, i.e., a WuR receiver (WuRx) and the main radio (for data transmission and reception), the network and Medium Access Control (MAC) layers should be modified to support the advantages of the WuR paradigm [4].

In this paper, we introduce a framework for M2WSNs based on the WuR paradigm and cross-layer design, 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 own information, the functions provided by its adjacent layers, and the information result-

* Corresponding author.
E-mail address: juan.aranda@usa.edu.co (J. Aranda).

https://doi.org/10.1016/j.adhoc.2020.102201
1570-8705/

ing from the cross-layer interactions. We have conducted experiments in hardware using real motes in an indoor testbed to validate the performance of the framework and to compare it against a single-radio architecture based on a LDC 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. To the best of our knowledge, the proposed framework is the first of its kind that integrates a multimodal approach regarding the data-reporting scheme and a cross-layer networking architecture based on the WuR paradigm, making it suitable for reliable and energy-efficient MultiModal WSNs. In summary, the main contributions of this paper are twofold:

- A reliable and energy-efficiency framework for M2WSNs is proposed that is based on the WuR paradigm and cross-layer design, suitable for monitoring oriented applications with low bandwidth requirements that operates simultaneously under normal circumstances and emergencies.
- A performance evaluation and comparison between the proposed framework and a traditional single-radio architecture are conducted in hardware.

The remainder of the paper is organized as follows: Section 2 provides a comparison of the proposed framework with the closest work presented in the literature. Next, Section 3 presents an overview of the framework design and its techniques is described in Section 4. Then, Section 5 gives an overview of the Cross-Layer Manager introduced to handle the interactions between the different layers included in the framework. Section 6 describes the experimental configuration. The results via hardware experiments are provided in Section 7. Finally, in Section 8, we present the conclusions of the paper and discuss future remarks.

2. Related work

Piyare et al. in [4] 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 in [5] proposed 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.

KRATOS and the proposed framework share common features such as the ContikiOS that allow us to employ ready-to-use modules such as the RIME stack [6] and CSMA protocol. But they differ in several aspects. First, the physical technology, KRATOS was proposed for LoRa implementations, and the proposed framework, for IEEE 802.15.4 deployments. Second, regarding MAC implementations, the authors in [5] did not provide enough details about it. They limited to mention that the RIME stack is used up to the top of the LoRa physical layer that usually is configured with a null-Mac as default MAC protocol. Finally, KRATOS offers an unmodified version of the collection tree protocol (CTP), available in the RIME stack, for path-selection and packet-forwarding procedures. The CTP version employed in the proposed framework has been modified to operate under a cross-layer design and a dual-radio architecture.

KRATOS has been used in [7] to validate a network architecture and on-demand time-division multiple access (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. The on-demand TDMA MAC scheme is offered for time synchronization and data collection. Compared to the proposed framework, this proposal presents several differences, such as the physical technology and communication approach. The proposed framework operates under the traditional IEEE 802.15.4 technology and employs a sender-initiated approach based on CSMA/CA (carrier sense multiple access with collision avoidance). A sender-initiated MAC scheme allows SNs to make decisions, in a distributed manner, without relying on their neighbors and the sink node. But this implies the implementation of a time network synchronization algorithm for data-collection and node scheduling without congesting the network while achieving high reliability and energy-savings. The framework implements an implicit pairwise time synchronization technique proposed in [8].

Kumberg et al. in [9] 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. Besides, T-ROME is executed in a distributed manner and under a sender-initiated communication approach. Finally, T-ROME supports functions in the network and link layers with non-cross-layer interactions between the application and physical layers, as considered in the proposed framework.

Bhuiyan et al. in [10] proposed e-sampling, an adaptive event-sensitive data acquisition and monitoring scheme for WSNs based on the WuR paradigm. e-sampling allows sensor nodes to dynamically adjust their sampling rates and perform an event-computation (i.e., make a decision about the absence or presence of an event) by analyzing the important frequency contents captured by their sensors. The sampling rate adjustment and event-computation are done in a decentralized manner. Before a node decides which data is reported to the sink, it wakes up its neighbors–those that include a WuRx hardware. Then, the nodes shares its event-computation, and performs a pairwise comparison. Finally, the node reports its decision to the sink. Once the sink receives the node's decision, it can request the event data stored into the node's memory in a centralized manner. The data dissemination procedure is done based on a sender-initiated approach using a basic routing technique over a TDMA MAC strategy.

Ait Aoudia et al. in [11] proposed a Star Network WuRx-MAC (SNW-MAC) protocol for time-driven data gathering in WSNs organized in a star-like topology. SNW-MAC provides asynchronous communications based on a receiver-initiated scheme and an id-based WuRx approach. However, SNW-MAC does not offer a complete framework with additional network techniques and interaction with upper layers. Besides, it might not be suitable for multi-hop and tree-like networks, commonly used in monitoring WSNs applications.

Recently, Sutton et al. in [12] 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 interferences, the authors implemented a distributed wake-up classifier using a decision-tree technique instead of an addressable or id-based wake-up scheme. BLITZ has been designed for event-driven networks compared to the proposed framework that operates in multimodal networks.

Table 1
Qualitative comparison of closest work.

Authors	Reporting Scheme	Networking Techniques	Wireless Technology	WuRx Scheme	Communication approach	Decision-making approach
Kumberg et al. [9]	Time-driven	T-ROME	IEEE 802.15.4	Id-based	Sender-Initiated	Distributed
Bhuiyan et al. [10]	Event-driven	Shortest Path/TDMA	IEEE 802.15.4	Broadcast-based	Sender-Initiated	Semi-Centralized
Ait Aoudia et al. [11]	Time-driven	None	IEEE 802.15.4	Id-based	Receiver-Initiated	Centralized
Piyare et al. [7]	Time-driven	RIME/On-Demand TDMA	LoRa	Id-based & Broadcast-based	Receiver-Initiated	Centralized
Sutton et al. [12]	Event-driven	CSMA	IEEE 802.15.4	Classifier	Sender-Initiated	Semi-Centralized
This work	**MultiModal**	2R-CTP/CSMA	IEEE 802.15.4	Broadcast-based	Sender-Initiated	Distributed

In Table 1, a qualitative comparison between the proposed framework and the closest work aforementioned is presented. All proposals employed a WuR approach in its design. Some following an id-based scheme (i.e., WuRx supports addressing), others a wake-up classifier, or a broadcast-based wake-up scheme. Different networking techniques were implemented at the network and link layers. The proposed framework implements an adapted version of CTP, available in the RIME stack of ContikiOS, based on the WuR paradigm and a sender-initiated communication approach. Besides, the framework provides a fully distributed and cross-layer approach for decision-making regarding the wake-up procedure and the data-reporting mode. These features allow SNs to make decisions without relying on their neighbors and the sink.

To the best of 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.

3. Framework design overview

In this section, we introduce a framework for M2WSNs suitable for monitoring oriented applications with low bandwidth requirements, that operates under normal circumstances and emergencies, using a dual-radio architecture based on the WuR paradigm. 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. Fig. 1 shows an overview of the proposed framework.

The left-hand side of Fig. 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 control and 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 Fig. 1 are presented in Table 2. i_m represents the data structure exchanged between the CLM and a particular layer.

In general, the proposed framework has the following features:

- **Responsive**. In emergencies, events and their associated data are propagated immediately towards the sink, throughout the network [12,13]. The M2WSNs do their 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 framework provides this feature mainly by the implementation of a multimodal switching mechanism and a wake-up protocol based on the WuR 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) [14].
- **Low bandwidth.** The framework is designed for applications with low bandwidth[1] (small packet and low transfer data rate) requirements, less than or equal to 250kbps that is supported by the IEEE 802.15.4 standard [15], 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 paper.

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.

[1] The term bandwidth refers to the data rate (bit per seconds).

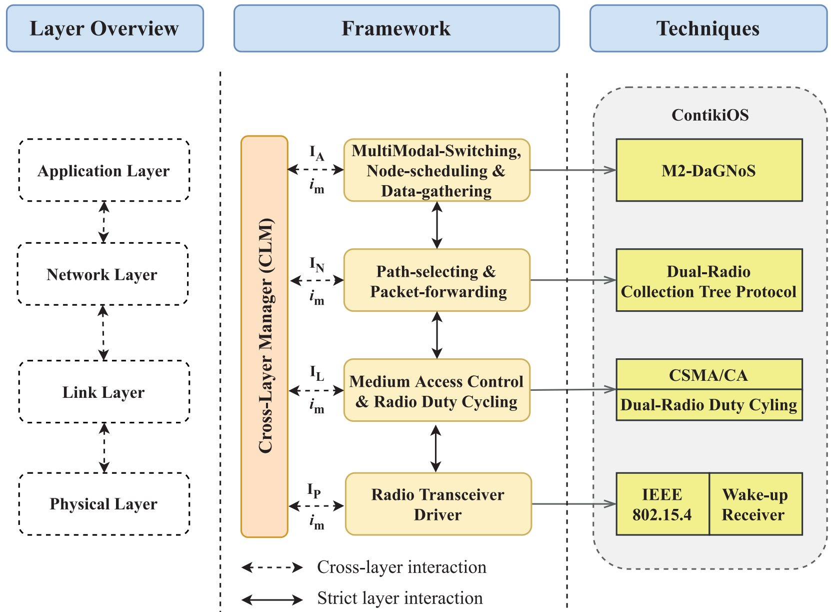


Fig. 1. A general framework for M2WSNs: an overview and its associated techniques..

Table 2
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.

4. Framework techniques

The right-hand side of Fig. 1 presents the techniques used to address the requirements, the middle diagram, requested by the framework.

4.1. Multimodal-switching, node-scheduling and data-gathering

The top module or application layer of the framework is in charge of performing the switching between time-driven (CMnt) and event-driven (EDR) data-gathering modes based on the circumstances presented within the area of supervising, i.e., under an emergency, the SNs employ the EDR capabilities, and during the period of calm of the event (i.e., under a normal situation), the SNs switch to the CMnt capabilities to periodically transmit up-to-date data of the current status of the supervising area to the sink (observer). This procedure is usually 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 [1].

We proposed in [16] an energy-efficient and distributed M2-DaGNoS technique for M2WSNs monitoring applications. M2-DaGNoS stands for **M**ulti**M**odal switching mechanism for **Da**ta **G**athering and **No**de **S**cheduling. 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 [1], 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 [17], combined with the parameter-event-detection algorithm described in [18], 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., 2.4 GHz). More details about the design, implementation, and performance evaluation of M2-DaGNoS is provided in [16].

4.2. 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 SNs 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–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 are them 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) [19].

The framework is implemented in ContikiOS, a lightweight and flexible operative system for low-power WSNs and Internet of Things (IoT) solutions [20]. ContikiOS offers a ready to use lightweight layered communication stack, known as RIME [6]. Within the RIME stack, there is an implementation of the standard CTP [19]. 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 [21].

Fig. 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. Several timers are used during the CTP operation that are 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 (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 [21].

$$rtmetric = \arg\min_{n\in N}\{rtmetric_n + ETX_n\} \tag{1}$$

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 Fig. 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. Fig. 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, 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 to collisions or ACK loss, the data packet is retransmitted every $T_{AW}/4$, as shown in Fig. 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.

4.3. Medium access control and radio duty cycling

The MAC 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 SNs 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 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 selected the well-known *Carrier-Sense Multiple Access with Collision Avoidance* (CSMA/CA) method supported by the standard IEEE 802.15.4 [15]. CSMA/CA is a widely used contention-based MAC protocol for wireless communications that does 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 implies additional packet re-transmissions at cost of an extra energy consumption. There is a CSMA/CA version for ContikiOS described in [22].

The 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 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 [23]. 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 2.4 GHz) 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 [5].

4.3.1. 2R-MAC operation from the WuRx side

The 2R-MAC protocol is only listening to the channel, awaiting to any data packet (i.e., wake-up signal (WuS)) to arrive and then, to wake the main radio (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 Fig. 4. 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.

4.3.2. 2R-MAC operation from the MR side

The 2R-MAC protocol follows a sender-initiated scheme, i.e., “a message source triggers the receiver to wake-up” [23]. Based on this scheme, the MR follows the state machine shown in Fig. 5. 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

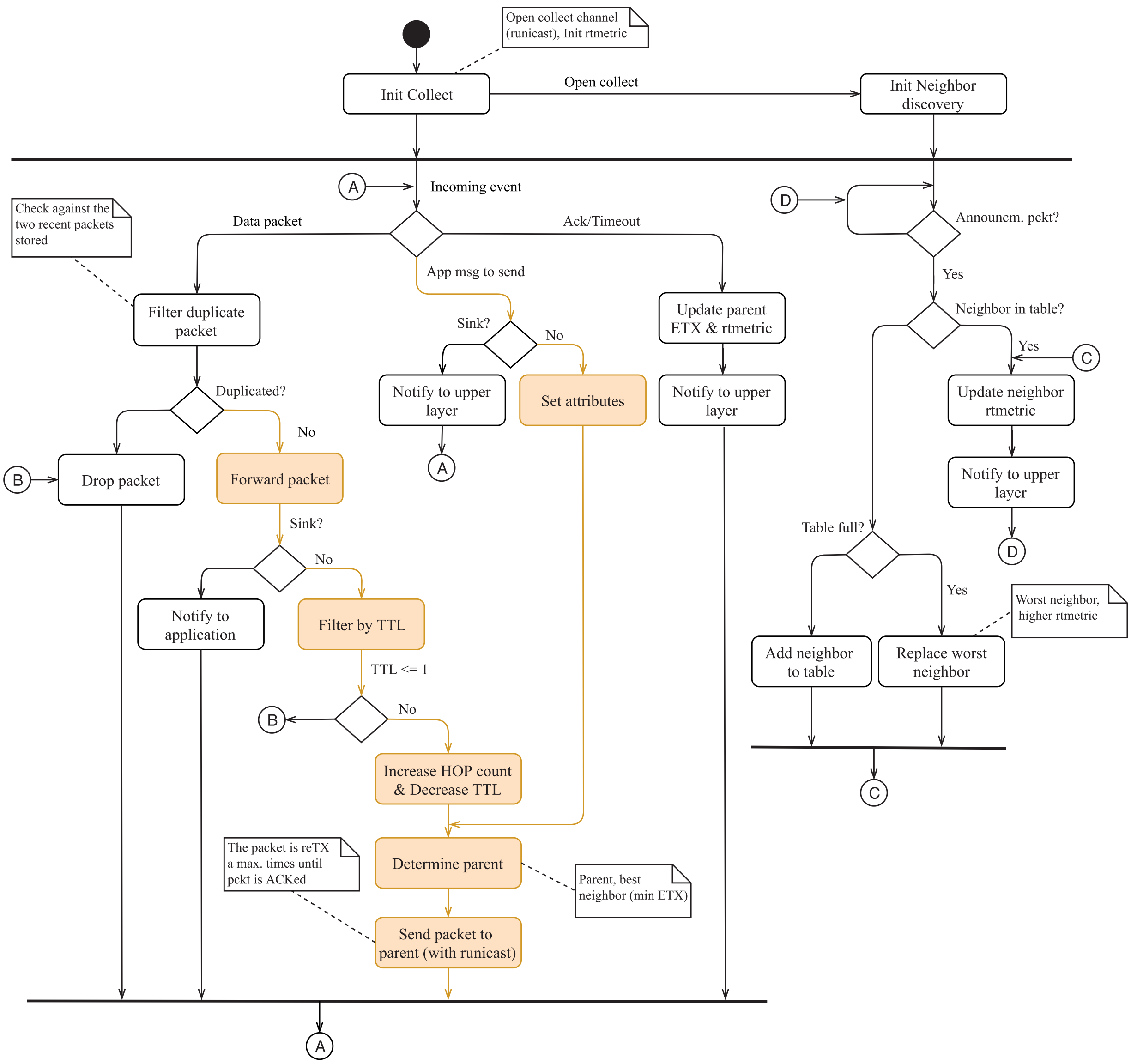


Fig. 2. The collect tree protocol operation implemented in ContikiOS (Adapted from [21])..

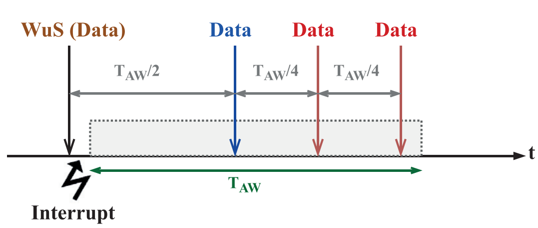


Fig. 3. Data packet retransmission time adjustment for a dual-radio scheme using CTP based on the WuR paradigm.

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 cross-layer manager 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.

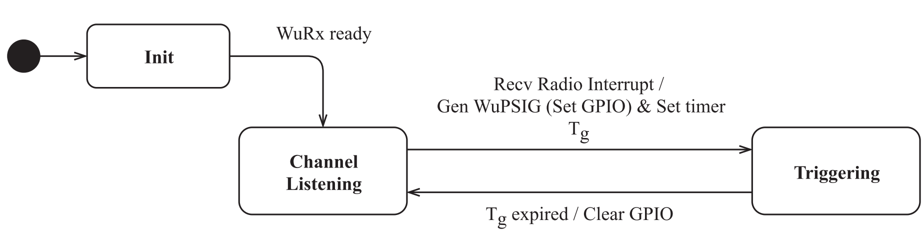


Fig. 4. WuRx application algorithm using 2R-MAC protocol..

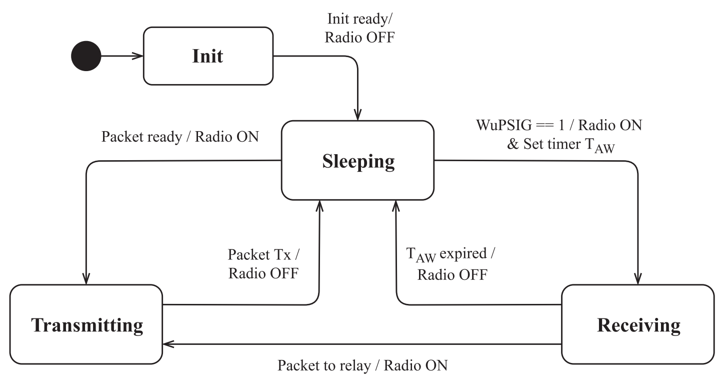


Fig. 5. Main node application algorithm using 2R-MAC protocol.

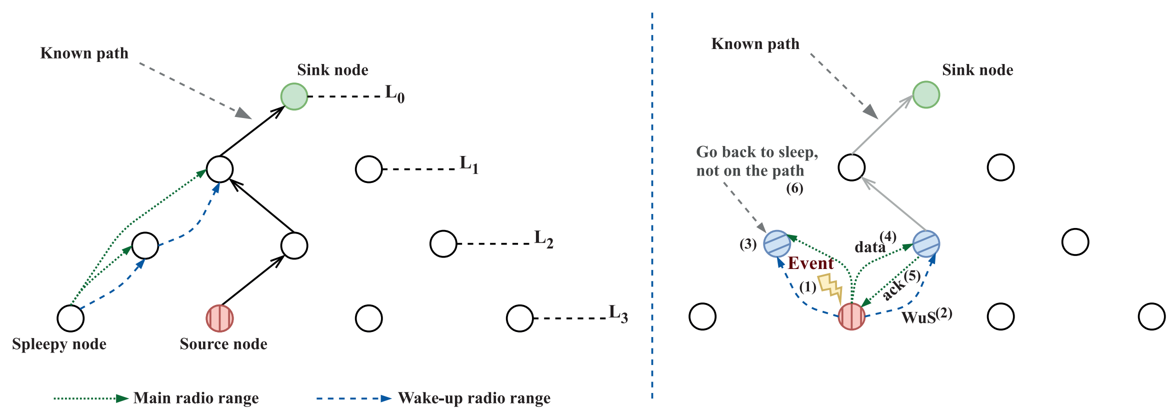


Fig. 6. A dual radio multi-hop communication schematic on a tree-like topology (Taken from [24]).

4.3.3. 2R-MAC and 2R-CTP cooperation for event data dissemination

When an event occurs within the M2WSNs, the SNs follows the procedure presented in Fig. 6 that describes a dual-radio multi-hop communication on a tree-like topology (left sketch), built and maintained by 2R-CTP. The right sketch shows a multi-hop operation of the wake-up protocol proposed in our previous work [24]. (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.

Our framework implements the “worst” scenario for wake-up and data dissemination toward the sink, as proposed in [24]: (i) non-addressing support is used (i.e., all SNs within the neighborhood are woken up to receive a packet (see Fig. 6)); (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 [9,12,25], regarding event reporting latency and energy savings.

4.4. 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 [15]. 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) [26]. 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 to mention that the WuR paradigm is started to be employed in WSNs as a solution for the LDC mechanism issues that usually waste energy, due the idle listening, and increase the reporting latency, due to the long-sleep intervals [7].

The WuRx driver is design to receive any data signal with a carrier frequency of 2.4 GHz 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

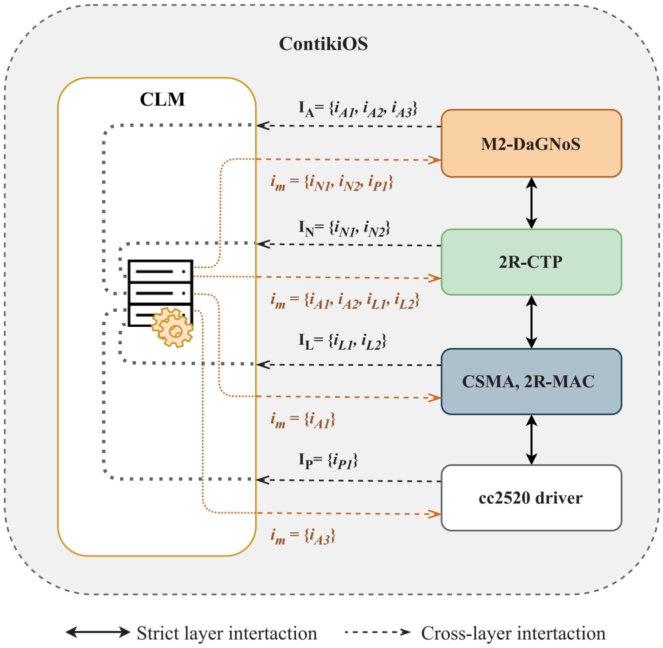


Fig. 7. Cross-layer interactions between modules of the proposed framework.

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.

5. Cross-layer manager

The framework follows a cross-layer design approach [14] 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 Fig. 1 serves as an interface between the higher layers and the physical layer. Fig. 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 2 shows the information shared with the CLM.

Fig. 7 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) 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.

6. Performance evaluation

In this section, we present the setup used for the performance evaluation of the proposed framework.

6.1. Experimental configuration

The entire framework has been implemented in a hardware prototype. The hardware prototype consists of pairwise connected radio-mote modules: one main radio and one WuRx connected by a hardware-software interface (i.e., via General-Purpose Input/Output (GPIO) ports), as shown in Fig. 8. The former for data transmission and reception procedures under the IEEE 802.15.4. The latter for wake-up signals reception at the 2.4 GHz frequency band.

During the time the experiments were performed, commercial WuRx hardware was not available on the market with the technical specifications of our implementation (e.g., in-band operation at 2.4 Ghz). Hence, we have designed and implemented a hardware prototype with independent power measurement that would then allow us to scale the results. The WuRx module consists of an FW-node (FW1), as shown in Fig. 8, with the functionalities of a real WuRx prototype, as proposed in the literature: (1) Be always listening to the communication channel for any WuS coming from other nodes in the M2WSNs. (2) If a WuS is received, the FW1 triggers an external interrupt signal (WuPSIG) from a GPIO to wake the main radio up (FW2).

The main radio-node (FW2) works as the primary node that remains in a sleep mode until some interrupt signal actives it via GPIO. The main radio has the following functions: (1) Receive a data packet from other SNs 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 the software perspective, the framework has been wholly implemented in ContikiOS, from the application layer to the physical layer, and the cross-layer manager entity. As shown in Fig. 9, both modules (i.e., FW1 and FW2) run a firmware under ContikiOS with the communication stack proposed. The main node supports CSMA/CA and 2R-MAC techniques at the link layer, the dual-radio collect tree protocol, 2R-CTP, at the network layer, and the multi-modal switching mechanism, M2-DaGNoS, 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. Finally, at the physical layer, lower-power RF chip drivers (e.g., cc2520) with IEEE 802.15.4 (at 2.4 GHz) is employed for data packet transmission and receiving procedures via the primary radio. Besides, a hardware interface is added to the ContikiOS for

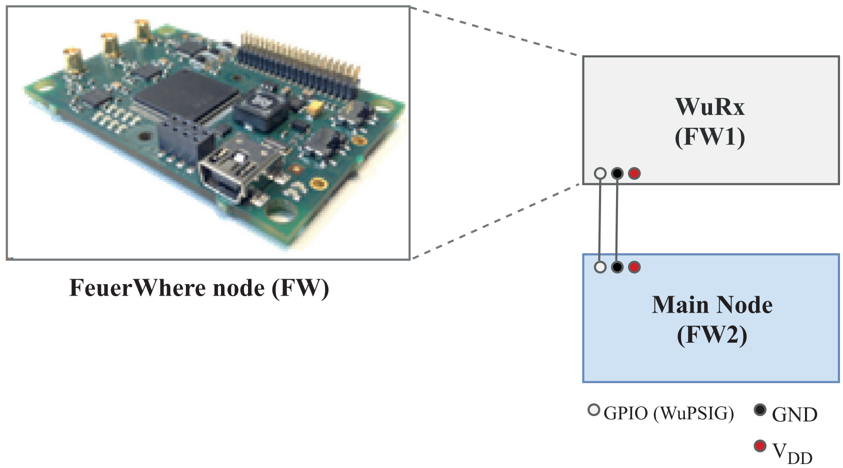


Fig. 8. Example of a dual-radio hardware architecture using two FW-nodes from IHP: One as WuRx (always-on scheme) and the other, as a main node (in sleep mode).

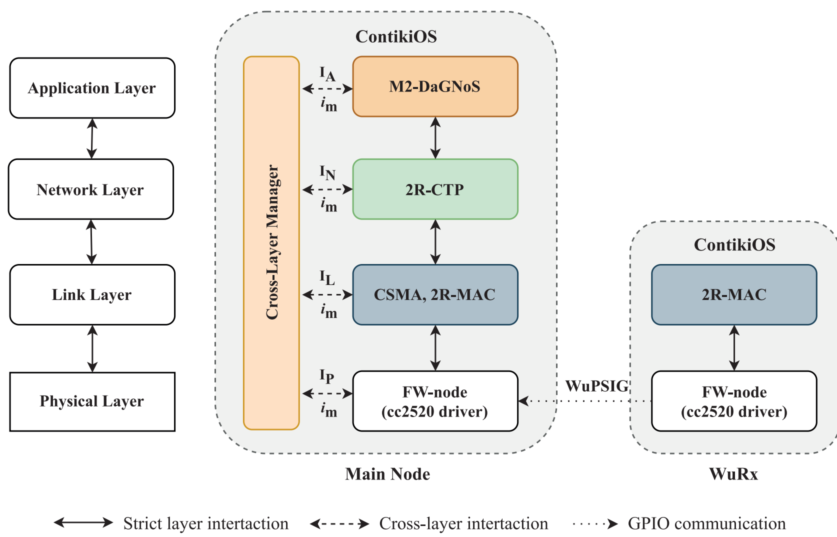


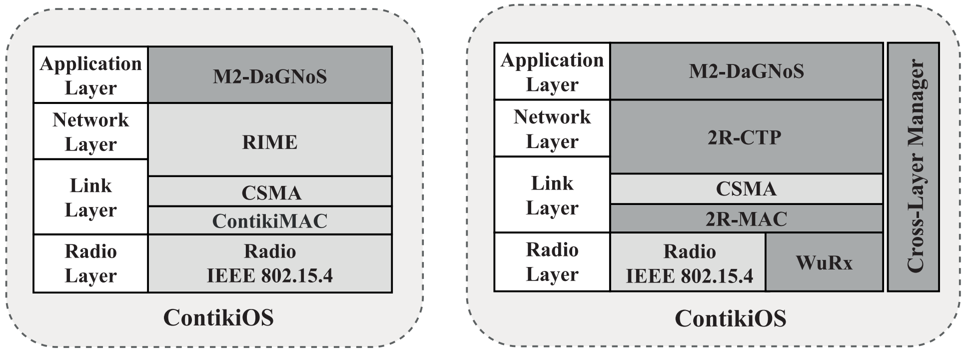
Fig. 9. A dual-radio architecture from a communication stack perspective following the structure of the proposed framework.

wake-up receiver interrupts handler (i.e., between the WuRx and the CPU of the main node using I/O ports (WuPSIG triggering)).

As mentioned in Section 5, the CLM module handles the non-adjacent interactions between the modules included in the framework (see Fig. 9–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 between application to network and application to link layers. These interactions might add additional features to the framework such as priority sending packet management during emergencies, dead nodes management, and dual switching between different RDC schemes, e.g., LDC and WuRx.

A priority packet management during emergencies might consists of providing to intermediate nodes the skill of identifying critical packets over regular packets such as the first packet associated with the occurrence of an event, so that, it is possible to speed up the propagation of critical packets toward the sink. One possible approach is to add priority tags into the packet header to prioritize critical packets over normal packets. Regarding the dead nodes management for continuous monitoring, the M2-DaGNoS mechanism implements a cyclical node sleep scheduling algorithm to extend the battery life of the nodes. However, in case a node starts running out of energy during a scheduling cycle, this node should have the capability to communicate this eventuality to the next supervisor node or to the entire neighborhood. This action might avoid or mitigate the problem of not monitoring the area due to dead nodes. Hence, one possible improvement is to include into the M2-DaGNoS mechanism, as a decision parameter, the node's residual energy (e.g., voltage) to decide when to communicate the eventuality of running out of energy. Consequently, it might be possible to reschedule the remaining nodes in the network to continue supervising the area.

We compare our framework against a single-radio architecture, as shown in Fig. 10(a). The single-radio architecture employs only one radio with IEEE 802.15.4 support when compared to our framework. As shown in Fig. 10(b), our framework 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., 2.4 GHz). At the link layer, the single-radio architecture implements ContikiMAC [2]–the tra-



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

Fig. 10. An overview of communication stacks implemented under different radio architectures. The dark gray modules are proposed for the framework.

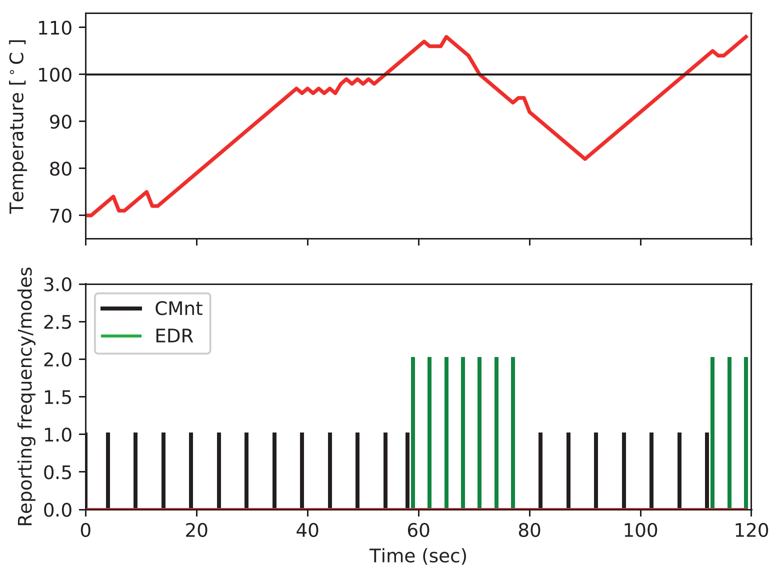


Fig. 11. Temperature profile with the reporting modes and frequencies for the validation process.

ditional asynchronous low-duty-cycling mechanism widely evaluated in the literature [25,27,28], and our framework implements 2R-MAC technique. 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 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 chose a scenario that considers both normal circumstances and emergencies. Under a normal circumstance, the working node is reporting to the sink every 5s (black lines), and during an emergency, every 2.5 s (green lines) within 120s. 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. Events are introduced into the network according to the temperature profile shown in Fig. 11. On Table 3, 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 on its main components: CPU, transceiver 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 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.

6.2. Proof-of-concept M2WSNs implementation

We deploy a proof-of-concept testbed in an indoor environment, as shown in Fig. 12, using ContikiOS and real motes, Feuer-

Table 3
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/5s	1 packet/5s
Packet rate (Event)	1 packet/2.5s	1 packet/2.5s
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)

Table 4
Summary of the main features of the FW-node (Based on [29]).

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

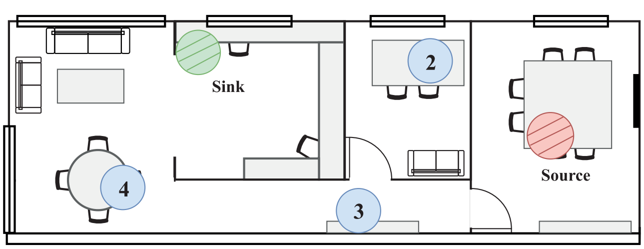


Fig. 12. Map of the indoor laboratory (area of 93m^2) used for the proof-of-concept.

Where (FW) nodes from IHP [29], within a well-defined topology, as shown in Fig. 13.

The indoor testbed setup in Fig. 12 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 m. Fig. 13 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 Fig. 6–based on a broadcast-based wake-up scheme.

Every node is implemented using a FW-node (Fig. 13). The framework is implemented in the hardware prototype proposed in Section 6.1. Each FW-node is powered with a 3.7V@1300mAh battery. The sink node is connected to a PC for data logging. A summary of the main features of the FW-node is provided in Table 4.

We have measured the power consumption with a Wireless Debugging and Power Measurement System–Wisdom platform–, proposed in [30], and a TI INA219 DC Current Shunt and Power Monitor, using the set-up shows in Fig. 14. 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 (e.g., 157mW to 2.2μW (-55dBm, 50m)[31]).

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 Fig. 15(a). For instances, in Fig. 15(b), 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.

7. Results

Fig. 16(a) shows the average power consumption of both architectures. The results indicate that our framework consumes slightly more energy (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. The WuRx module (green bar) consumes 3 orders of magnitude less than the main radio (red bar).

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 [2], but at the cost of higher latency in multi-hop communications (see Fig. 16(b)). 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 2.4 GHz makes the node 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 continuously switching on and off the

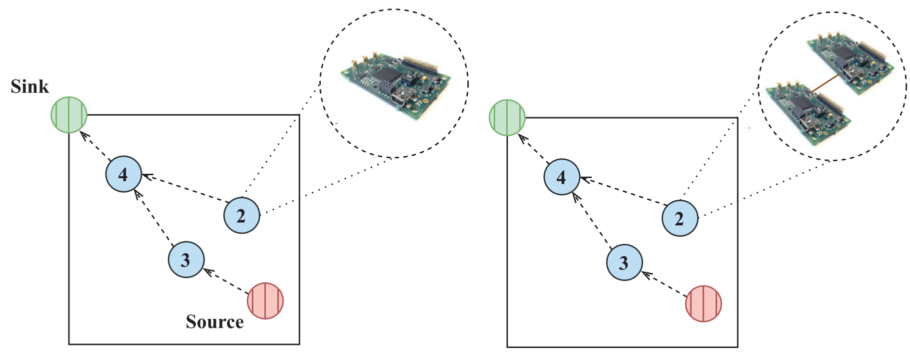


Fig. 13. Logical topology configured for the proof-of-concept to compare the single-radio architecture (left graph) and our framework (right graph).

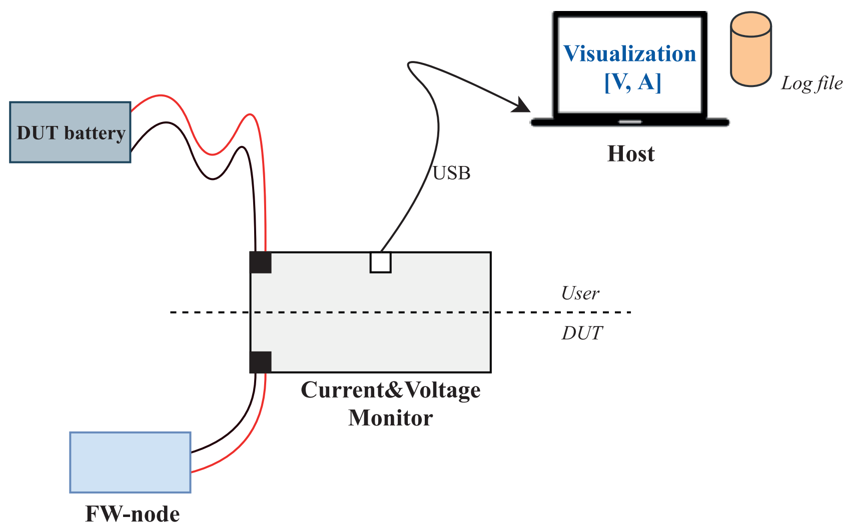
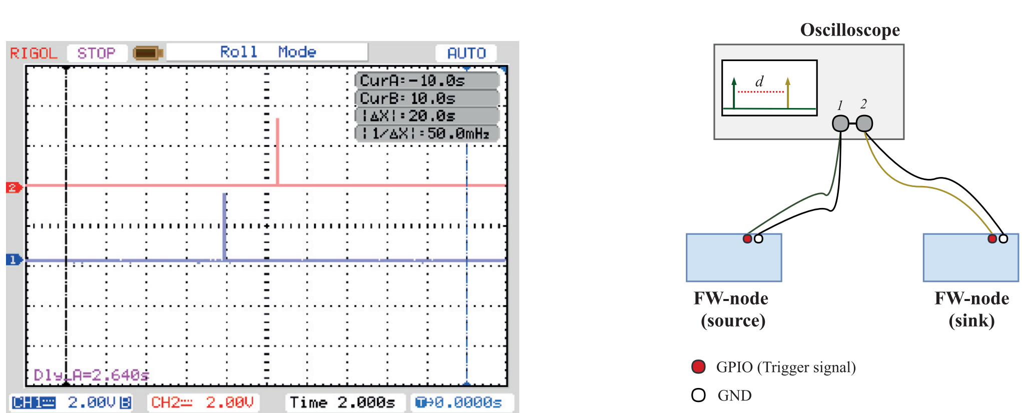


Fig. 14. Power consumption measurement set-up.



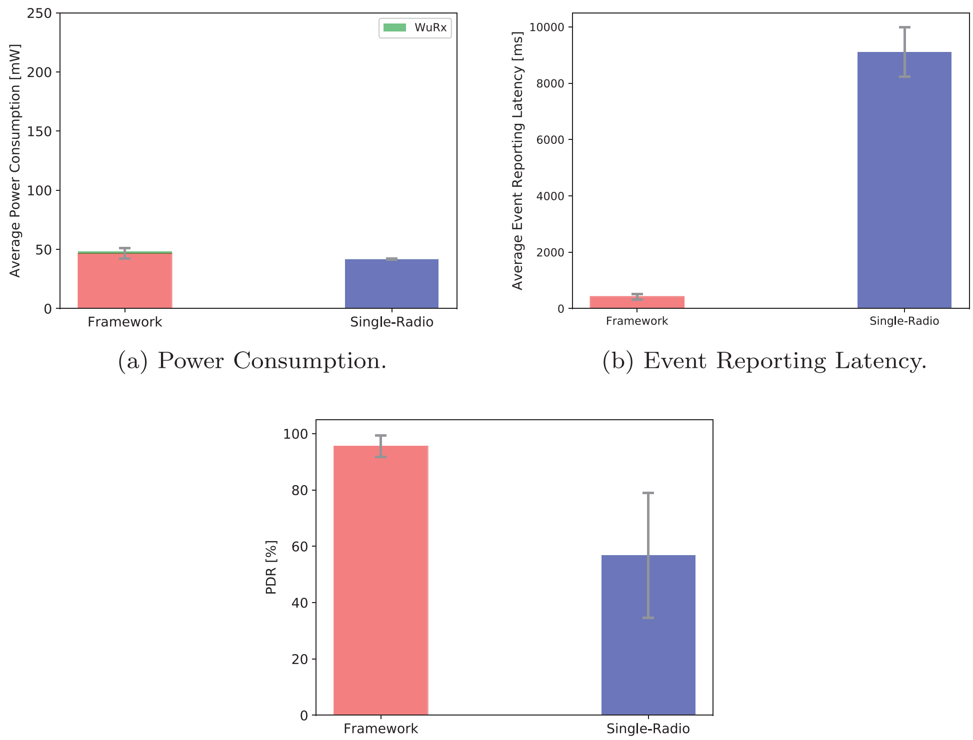
(a) Trigger signals. Source (blue signal), and Sink (red signal).

(b) Set-up diagram for latency measuring.

Fig. 15. Event reporting latency measurement using trigger signals and an oscilloscope.

main 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 [12], or the data packet is transmitted to the sink within one-hop [25] or in few hops compare to the number of hops in the woken-up path [9]. Besides, the data packet [12] or WuS [25] might not be acknowledged. Finally, these approaches considered addressing support in their WuRx implementation, designed for event-driven networks. As future work, by considering some of



(a) Power Consumption.

(b) Event Reporting Latency.

(c) Packet Delivery Ratio.

Fig. 16. Performance evaluation comparison between a single-radio architecture using a LDC scheme and our framework based on a wake-radio approach.

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 architectures. The results in Fig. 16(b) 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 Fig. 16(b)–436.18ms for our framework, and 9.11s for the single-radio architecture.

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, under ideal conditions, to forward a packet or to stop emitting the full data packet until the designated receiver mote wakes up and send an acknowledgment. Under the effect of a real channel (over the packet transmisssion, 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 [25]. 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 an 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 5–2.5s (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 buffer overload and congestion situations in the multi-hop network, affecting the latency performance–due to packet loss [32]. 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 [33,34].

Fig. 16(c) 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, congestion situations 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[2], but at the cost of a higher latency and energy consumption [25], due to the arise of extra congestion and collision issues.

[2] The maximum CSMA retransmissions for the single-radio architecture have been configured using the default value in ContikiOS (see Table 3). 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.

8. Conclusions and future work

In this paper, we presented an reliable and energy-efficiency framework for M2WSNs based on the WuR paradigm and Cross-Layer design, 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 own information, the functions provided by its adjacent layers, and the information resulting from the cross-layer interactions.

We have reported preliminaries experimental validation results using real motes in an indoor testbed that validate the performance of the framework against a single-radio architecture based on an LDC technique. The framework has shown 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.

The framework has been evaluated in a small indoor testbed. We can expect when the network size (density of nodes per area) increases, the network performance will be affected. For instance, for large-scale M2WSNs based on tree-like topology, the deep of the tree might grow as more nodes are added. Consequently, longer multi-paths toward the sink might be established to propagate an event packet, resulting in higher end-to-end latency [25,33]. Besides, for a high density of nodes per area, the possibilities of collisions, longer CSMA back-off time, retransmissions and congestion situations might also increase, affecting the latency results [25]. Similar behavior is expected for the power consumption performance [24]. Moreover, as the density of nodes per area increases, the false wake-up cases within bigger neighborhoods might also be raised, specialty, when a broadcast-based wake-up scheme is considered, contributing to the total power consumption of the network.

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 devices for simultaneous measurement of power consumption and latency. Besides, this platform will allow us to replicate the results in a controlled manner and make strong conclusions about the performance of the network under study.

We also propose to explore additional features such as priority sending packet management during emergencies, dead nodes management for continuous monitoring, and dual switching between different radio duty-cycling schemes using a cross-layer design.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the cooperation of all partners within the *Centro de Excelencia y Apropiación en Internet de las Cosas (CEA-IoT)* and the *IHP – Innovations for High Performance Microelectronics*. The authors would also like to thank all the institutions that supported this work: the Colombian Ministry for the Information and Communications Technology (*Ministerio de Tecnologías de la Información y las Comunicaciones - MinTIC*), the Colombian Administrative Department of Science, Technology and Innovation (*Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias*) through the *Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación Francisco José de Caldas* (Project ID: FP44842-502-2015), the German Research Foundation (Deutsche Forschungsgemeinschaft - DFG) (Project ID: KR 3576/21-1), and the *Universidad Sergio Arboleda* (Project ID: IN.BG.086.17.007).

References

[1] J. Aranda, D. Mendez, H. Carrillo, MultiModal wireless sensor networks for monitoring applications: a review, J. Circ. Syst. Comput. 29(2). (n.d.), doi:10.1142/S0218126620300032.
[2] A. Dunkels, The ContikiMAC Radio Duty Cycling Protocol, 2011, pp. 1–11. SICS Technical Report T2011:13, ISSN 1100-3154
[3] A. Elgani, M. Magno, F. Renzini, L. Perilli, E.F. Scarselli, A. Gnudi, R. Canegallo, G. Ricotti, L. Benini*, Nanowatt wake-up radios: discrete-components and integrated architectures, in: 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2018, pp. 793–796, doi:10.1109/ICECS.2018.8617961.
[4] R. Piyare, A.L. Murphy, C. Kiraly, P. Tosato, D. Brunelli, Ultra low power wake-up radios: a hardware and networking survey, IEEE Commun. Surv. Tut. PP (99) (2017) 1, doi:10.1109/COMST.2017.2728092.
[5] R. Piyare, A.L. Murphy, M. Magno, L. Benini, KRATOS: an open source hardware-software platform for rapid research in LPWANs, in: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2018, pp. 1–4, doi:10.1109/WiMOB.2018.8589171.
[6] A. Dunkels, Rime — a lightweight layered communication stack for sensor networks, in: Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session, 2007. Delft, The Netherlands.
[7] R. Piyare, A.L. Murphy, M. Magno, L. Benini, On-demand LoRa: asynchronous TDMA for energy efficient and low latency communication in IoT, Sensors 18 (11) (2018), doi:10.3390/s18113718.
[8] S. Chen, A. Dunkels, F. Osterlind, T. Voigt, M. 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.
[9] T. Kumberg, M. Schink, L.M. Reindl, C. Schindelhauer, T-ROME: a simple and energy efficient tree routing protocol for low-power wake-up receivers, Ad Hoc Netw. 59 (2017) 97–115, doi:10.1016/j.adhoc.2017.02.003.
[10] M.Z.A. Bhuiyan, J. Wu, G. Wang, T. Wang, M.M. Hassan, E-Sampling: event-Sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems, ACM Trans. Auton. Adapt. Syst. 12 (1) (2017) 1–29.
[11] F. Ait Aoudia, M. Gautier, M. Magno, O. Berder, L. Benini, SNW-MAC: an asynchronous protocol leveraging wake-up receivers for data gathering in star networks, in: Sensor Systems and Software, Springer International Publishing, 2017, pp. 3–14.
[12] F. Sutton, R.D. Forno, J. Beutel, L. Thiele, BLITZ: low latency and energy-efficient communication for event-triggered wireless sensing systems, ACM Trans. Sen. Netw. 15 (2) (2019) 25:1–25:38, doi:10.1145/3309702.
[13] J. Aranda, H. Carrillo, D. Mendez, Enhanced multimodal switching mechanisms for node scheduling and data gathering in wireless sensor networks, in: IEEE Colombian Conference on Communications and Computing, 2017, pp. 1–6.
[14] R. Ranjan, S. Varma, Challenges and implementation on cross layer design for wireless sensor networks, Wirel. Personal Commun. 86 (2) (2016) 1037–1060.
[15] IEEE Standard for Low-Rate Wireless Networks, 2016, pp. 1–709. IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)
[16] J. Aranda, D. Mendez, H. Carrillo, M. Schölzel, M2-DaGNoS: a data gathering and node scheduling mechanism for multimodal wireless sensor networks, Comput.Commun. 148 (2019) 165–175, doi:10.1016/j.comcom.2019.09.012.
[17] Y. Kang, B. Hu, Y. Ding, J. Tan, A hybrid node scheduling approach based on energy efficient chain routing for WSN, Adv. Mech. Eng. 2014 (2014) 1–12.
[18] B. Lee, K. Lim, An energy-efficient hybrid data-gathering protocol based on the dynamic switching of reporting schemes in wireless sensor networks, IEEE Syst. J. 6 (3) (2012) 378–387, doi:10.1109/JSYST.2011.2171269.
[19] G. Omprakash, F. Rodrigo, J. Kyle, M. David, L. Philip, Collection tree protocol, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '09, ACM, 2009, pp. 1–14, doi:10.1145/1644038.1644040.
[20] A. Dunkels, B. Gronvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: 29th Annual IEEE International Conference on Local Computer Networks, 2004, pp. 455–462, doi:10.1109/LCN.2004.38.
[21] W. Van Heddeghem, Cross-layer Link Estimation For Contiki-based Wireless Sensor Networks, 2009 Master's thesis.
[22] H. Tall, G. Chalhoub, M. Misson, Implementation and performance evaluation of IEEE 802.15.4 unslotted CSMA/CA protocol on Contiki OS, Ann. Telecommun. 71 (9) (2016) 517–526, doi:10.1007/s12243-016-0522-y.
[23] R. Piyare, T. Istomin, A.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, Junction Publishing, USA, 2017, pp. 48–53.
[24] J. Aranda, M. Schölzel, D. Mendez, H. Carrillo, Multimodal wireless sensor networks based on wake-up radio receivers: an analytical model for energy consumption, Rev. Facult. Ingeniería Univ. Antioquia (91) (2019) 113–127.
[25] S.L. Sampayo, J. Montavont, F. Prgaldiny, T. Nol, Is wake-up radio the ultimate solution to the latency-energy tradeoff in multi-hop wireless sensor networks? in: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2018, pp. 1–8.

[26] K. Roussel, Y.-Q. Song, A Critical Analysis of Contiki's Network Stack for Integrating New MAC Protocols, 2013, p. 13. https://hal.inria.fr/hal-01202542.

[27] M. Michel, B. Quoitin, Technical Report : ContikiMAC vs X-MAC Performance analysis, 2015, pp. 1–28.

[28] K. Roussel, Y.-Q. Song, O. Zendra, Lessons Learned through Implementation and Performance Comparison of Two MAC/RDC Protocols on Different WSN OS, 2015, pp. 1–25. INRIA Nancy. RR-8777 hal-01202664

[29] K. Piotrowski, S. Ortmann, P. Langendörfer, Multi-radio wireless sensor node for mobile biomedical monitoring, in: Biomedical Engineering / Biomedizinische Technik, 57, 2012, pp. 725–728.

[30] M. Frohberg, E. Batlkhagva, M. Schölzel, WISDOM a wireless debugging and power measurement system for field tests and device observation in WSN, in: 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2017, pp. 319–324.

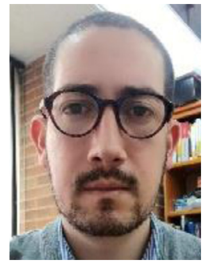
[31] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, L. Benini, Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-Up radio receiver, IEEE Trans. Ind. Inf. 12 (2) (2016) 633–644, doi:10.1109/TII.2016.2524982.

[32] O. Chughtai, N. Badruddin, M. Rehan, A. Khan, Congestion detection and alleviation in multihop wireless sensor networks, Wirel. Commun. Mobile Comput. 2017 (2017) 1–13, doi:10.1155/2017/9243019.

[33] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G.U. Gamm, L. Reindl, Has time come to switch from duty-cycled MAC protocols to wake-up radio for wireless sensor networks? IEEE/ACM Trans. Netw. 24 (2) (2016) 674–687, doi:10.1109/TNET.2014.2387314.

[34] M. Bezunartea, M. Banh, M. Gamallo, J. Tiberghien, K. 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, in: ICC 17, Association for Computing Machinery, 2017, doi:10.1145/3018896.3018935.

Juan Aranda received a B.S. degree in Electronic Engineering and a M.S. degree in Electronics Engineering from Universidad de los Andes, Bogotá, Colombia, in 2010 and 2012, respectively. He is currently pursuing a Ph.D. at Pontificia Universidad Javeriana, Bogotá within the CEA-IoT. Since 2014, he has been an Associate Professor with Escuela de Ciencias Exactas e Ingeniería, Universidad Sergio Arboleda, Bogotá. His current research interests are wireless sensor networks and connectivity on Internet of Things (IoT) applications.



Diego Méndez Chaves is an Associate Professor in the Department of Electronics Engineering at the Pontificia Universidad Javeriana, Bogotá, Colombia. He received his Ph.D. (2012) and his M.Sc. (2011) in Computer Science from the University of South Florida, Tampa FL, USA, his M.E. (2008) from the Universidad de Los Andes, Bogotá, Colombia, and his B.E (2005) in Electronics Engineering from the Universidad Nacional, Bogotá, Colombia. Diego's research interests include Internet of Things (IoT), embedded systems, wireless sensor networks, participatory sensing, digital systems design, operating systems and high-level systems design.

Henry Carrillo received the B.Eng. degree in Electronics Engineering from Universidad del Bogotá Norte, Barranquilla, Colombia, in 2007; the M.Sc. degree in Electronics Engineering from Pontificia Universidad Javeriana, Bogotá, Colombia, in 2010; and the Ph.D. degree in Computer Science from Universidad de Zaragoza, Zaragoza, Spain, in 2014. He is professor with Escuela de Ciencias Exactas e Ingeniería, Univesidad Sergio Arboleda, Bogotá, Colombia. His research interests include the problems of mapping and localization for autonomous systems.

Mario Schölzel received a diploma degree in computer science in 2001 from the Brandenburg University of Technology (BTU) in Cottbus. From 2001 to 2006 he worked in the compiler construction group at BTU and received its doctoral degree in 2006. Starting in 2007, he worked in the computer engineering group at BTU in the field of fault tolerant computing. In 2014 he has finished his habilitation thesis. Since 2018 he is Chair for Processors and Digital Systems at Department for Informatics, Automation and Electronics (IAE) at the University of Applied Sciences in Nordhausen, and since 2014 he is scientist at the IHP in Frankfurt (Oder). His research interest includes dependable and fault tolerant embedded systems, including sensor networks.