



HAL
open science

Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?

Sebastian Lucas Sampayo, Julien Montavont, Fabien Pregaldiny, Thomas Noel

► To cite this version:

Sebastian Lucas Sampayo, Julien Montavont, Fabien Pregaldiny, Thomas Noel. Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct 2018, Limassol, Cyprus. 10.1109/WiMOB.2018.8589163 . hal-03698357

HAL Id: hal-03698357

<https://hal.science/hal-03698357v1>

Submitted on 17 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?

Sebastian L. Sampayo, Julien Montavont, Fabien Prégaldiny, Thomas Noël
ICube Laboratory, University of Strasbourg, France
{sampayo, montavont, f.pregaldiny, noel}@unistra.fr

Abstract—In Wireless Sensor Networks (WSN), duty-cycled Medium Access Control (MAC) protocols trade off latency for energy efficient operation. Over the past few years, Wake-Up Radio (WuR) has been presented as the ultimate solution for this tradeoff, allowing to reduce both at the same time. However, this might not be the general case regarding the large range of network configurations used in WSN. Several previous works have been done comparing WuR to traditional duty-cycled solutions, but no one has investigated before the limitations of this technology. In this article, we analyze the benefits and drawbacks of using WuR in multi-hop WSN. We also identify black spots in WuR that have not been investigated yet. Our study is based on evaluations using COOJA, a simulator for networks of ContikiOS nodes. A traditional duty-cycled MAC protocol is also included in our study for comparative purposes. From our study, we quantify the performances of WuR and provide some guidelines on how this technology can be efficiently used in multi-hop wireless sensor networks.

Index Terms—WSN, Wake-Up Radio, analysis, simulation, MAC protocols, asynchronous communications

I. INTRODUCTION

Self-driving cars, smart cities and health monitoring are applications of the Internet of Things (IoT) supporting autonomous and intelligent decision making systems. Many times physical sensors are connected together wirelessly building Wireless Sensor Networks (WSN). These devices are generally low cost and resource-constrained.

Traditionally, energy consumption is reduced in those networks with the help of duty-cycled Medium Access Control (MAC) protocols trading off latency for energy efficient operation [1]. However, such solutions still suffer from various limitations such as idle listening and overhearing that significantly affects the energy consumption. Wake-Up Radios (WuR) have advanced in recent years as a definitive and general solution to the energy consumption and latency tradeoff in multi-hop WSN [2]. WuR idea is simple and the behavior studied in this work is described in Section II. A complete description of the state-of-the-art of this technology is provided in [2]. In a few words, WuR adds a secondary ultra-low-power radio (the Wake-Up Radio) that stays on all the time, while the high power main radio (MR) stays sleeping. When a source node wants to initiate the communication it sends a WuR packet, called Wake-Up Signal (WuS), towards the destination. Upon

reception, the destination issues an interrupt to wake up the MR. Finally, the source and the destination can exchange data and acknowledgment through their MR. This way the idle listening is reduced because the MR is sleeping the whole time and wakes up asynchronously. Although the WuR is always-on listening, its power consumption is around 4 orders of magnitude less [2] than that of the MR [3]. In addition, the WuS can contain an address to identify the destination node so that it is the only one that wakes up. This way it also avoids overhearing, as the MR stays sleeping if the node is not the intended destination. However, the drawback of the ultra-low power consumption is that the sensitivity is dramatically decreased. This means that if the sender transmits at the nominal power, then the range of the WuR is shorter than that of the MR. As a result, waking up a destination may require to relay the WuS through intermediate nodes, potentially increasing the end-to-end latency. In addition, the hardware design of the WuR and the modulation used are very simple [2], which makes us wonder how robust it is against interferences or collisions.

Remarkable comparative analysis with duty-cycled MAC protocols have been done in the literature, including [4] and [5]. A review of those previous works is presented in Section V. However, to the best of our knowledge, there has not been any study on the problems cited above. In this article, we will investigate the benefits, drawbacks and tradeoffs of using WuR in multi-hop WSN in terms of the packet delivery ratio, latency and power consumption. Our analysis is based on evaluations using COOJA, a simulator for networks of ContikiOS nodes. The setup used throughout this study is described in Section III.

Our findings presented in Section IV show that there is a threshold in the size of the network for WuR to perform efficiently. Increasing the network size beyond this threshold significantly degrades the WuR performance, making a traditional duty-cycled MAC protocol a better choice for such configuration. We also show that acknowledging the WuS is problematic in the presence of collisions, because it decreases seriously the reliability of the network. Finally, our conclusions and future work are presented in Section VI.

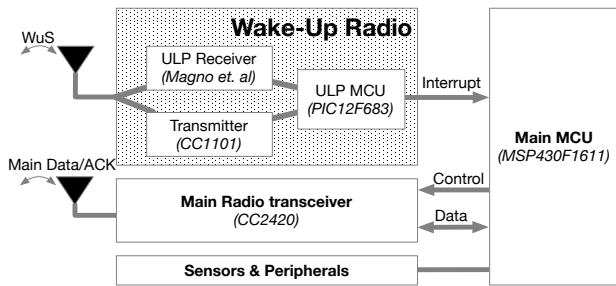


Fig. 1. Wake-Up Radio system architecture

II. WAKE-UP RADIO FEATURES

A. Naming convention

We will use the following terms throughout this article:

- Scenario: a fixed number of nodes running a specific protocol in a fixed medium model
- WuS Request (WuS REQ): WuS sent from the source to the destination
- WuS Acknowledgment (WuS ACK): WuS sent from the destination to the source
- WuS path: The path of nodes that the WuS has to travel to reach its destination
- WuS sequence: The sequence of retransmissions of WuS that needs to be made to travel the path
- Main data: Main data packets sent over the MR
- Main sender: The source node of the communication, the one who desires to send a data packet
- Main destination: The destination node of the communication, the one who should receive the data packet. It is also the final destination of the WuS path

B. Wake-up radio systems

The idea behind Wake-Up Radios is to reduce idle-listening, overhearing and latency. To do so, a secondary ultra-low-power radio (WuR) is added to the node, as shown in Fig. 1.

This WuR is always on, listening for WuS. Before sending a data packet via the MR, a source node sends a WuS towards the destination. The reception of WuS triggers the receiving node to wake up its MR, now waiting for the reception of the pending data packet. Upon data and acknowledgment exchange, the destination puts back its MR into sleep in order to save energy.

The modulation used commonly to send WuS is On-Off-Keying (OOK) because of the easiness of its implementation in hardware. Essentially, the WuR block consists of a receiver, a transmitter and an optional module to decode the WuS. Both the transmitter and the decoding module are common chips (such as CC1101 or CC2420). An ultra-low power MCU, like the PIC12F683, is normally used to decode the WuS and communicate with the main node's MCU through SPI. The latter module is sometimes not used so that the WuS is simpler, but also wakes up every node in the neighborhood. If this module is not used, nodes that are not intended to participate

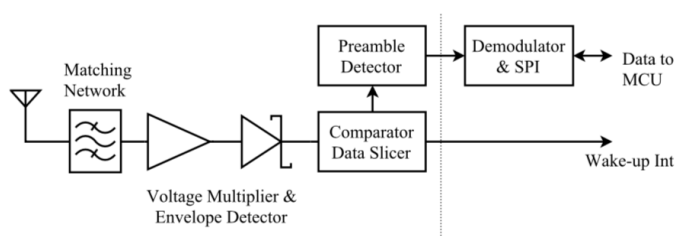


Fig. 2. Wake-Up Radio receiver blocks [2], [6]

in the communication wake up their main receiver and listen to the medium wasting energy. This is called in the literature a *false-positive*.

The true novelty in technology of the WuRs lies in the receiver. The block diagram of the circuit is depicted in Fig. 2. Although the implementation is a very simple circuit, there are not too many off-the-shelf modules available in the market for its application to WSN nodes. A clear disadvantage of these receivers is that the sensitivity achieved is lower than that of the main node's transceiver. This leads to a mismatch between the WuR's and the MR's range. As a result, if a destination is reachable via the MR but not via WuR, the WuS should be relayed by intermediate nodes before reaching the destination. To overcome such a situation, we can increase the transmission power of the WuS with a power amplifier, but this leads to a higher power consumption. In addition, regulatory requirements may not allow for such high output powers [4].

Throughout this work, we use the most common architecture with the decoding module that we have just described, and we use the nominal output power of the commercial transmitters, leading to the range mismatch problem. For more examples and variations of the WuR architecture, the reader is referred to [2].

C. Single WuS

In this article, we begin with a simple scenario where we have 4 nodes in a row. Main sender (node 4) and destination (node 1) are placed at the opposite borders of the line-shaped network. The main radio range is large enough so that both are connected through a single hop. However, the WuR range is small enough so that only adjacent nodes are connected. That means that for 4 to wake up 1 it needs to send a WuS to 3, and 3 needs to retransmit it to 2, and finally, 2 needs to retransmit it to 1. This operation is illustrated in Fig. 3. Such a situation should be very common in real life deployments due to the range mismatch between MR and WuR. However, analyzing the impact of relaying WuS has never been investigated before.

A short time after the WuS was sent, the main sender transmits the data packet over the main radio. From now on we define this short amount of time as the *sync delay* because node 4 is waiting for node 1 to wake up. After the data is received by the destination, an acknowledgment is sent back to the sender. Fig. 4 illustrates this single mode of operation. Notice that a longer WuS path makes the destination wake up later. So if the sync delay is not long enough the data will be

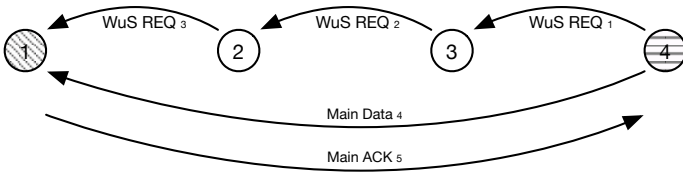


Fig. 3. Single WuS protocol

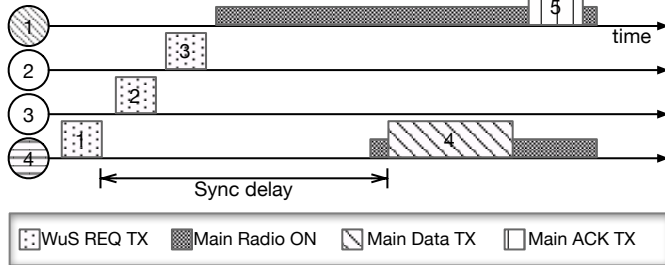


Fig. 4. Single WuS protocol timeline

transmitted when the destination is still sleeping, requiring a retransmission of the packet.

D. WuS ACK

The usage of WuS Acknowledgments (WuS ACKs) is not clearly pointed out in the literature. It is in some cases used as a response of a potential next hop in reactive protocols [7], but there is no indication on an end-to-end acknowledgment of the WuS through several hops. This way it would be possible to avoid the retransmissions problem related to a long WuS path and late destination wake up. For this, we introduce WuS ACK packets. The WuS ACK is sent by the main destination when it receives a WuS REQ and wakes up. This type of packet is relayed in the same fashion as the WuS REQ but in the opposite direction, towards the main sender. Upon reception of the WuS ACK, the sender is assured that the final destination is ready to receive the pending packet. This means that, in our example, firstly there is a WuS sequence from node 4 to node 1, secondly a WuS ACK sequence from node 1 to node 4, and only after both are successful, the main data and data ACK are exchanged between main node 4 and node 1. This scenario is illustrated in Fig. 5.

III. SIMULATION FRAMEWORK

There are lots of simulators and simulation frameworks for WSN in general [8] [9], but not so much has been done in the area of multiple radios. Several authors have been working with GreenCastalia [10], an extension of OMNet++ [8], that allows to easily model and simulate networks of embedded devices with energy-harvesting capabilities. However, this software cannot reproduce exactly the firmware that runs on the real devices and it does not consider the entire stack. In this work, we used WaCo [11], a COOJA extension that models a WuR prototype proposed in [6], and implements a WuR MAC protocol called W-MAC. COOJA is a software that simulates

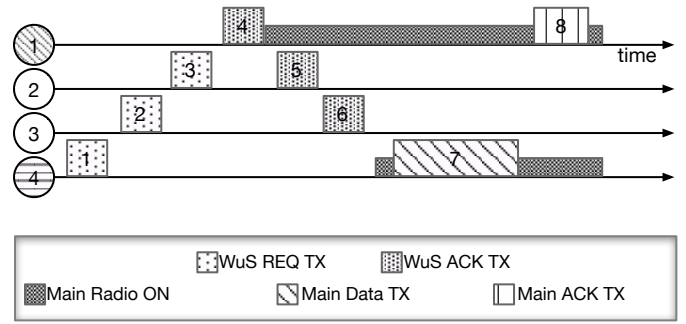


Fig. 5. WuSACK protocol timeline

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation duration	120 s
Repetitions of each simulation	6
ContikiMAC channel check period	125 ms
MAC layer	CSMA (Contiki version)
Max CSMA retransmissions	7
Network layer	RIME [12]
Packet rate	1 packet/s
WuS packet length	16 bits
WuS data rate	100 kbps
Main data packet length	43 bytes
Main data ACK packet length	5 bytes
Main radio data rate	250 kbps
Main Node	Sky mote [3]
WuR HW prototype	[6]
WuR Supply Voltage	1.8 V
WuR TX current	8 mA
WuR RX current	2 mA
WuR idle listening power consumption	1.944 μ W
Main Radio medium model	UDGMConstantLoss
WuR Radio medium model	UDGMConstantLoss
Main radio RX success ratio	100%, 80%
WuR RX success ratio	100%, 90%, 80%
WuR Sync delay	1.8ms, 3.1 ms, 6.45 ms

a network of ContikiOS nodes, allowing us to easily port the code for real devices. WaCo's performance was validated against experiments with the real prototype [11]. Although this software provides the basic elements to handle the WuR, there is no support for relaying WuS packets or changing the RX success probability of the WuR medium. These features have been added to WaCo and the W-MAC layer in ContikiOS in this contribution. Note that WaCo supports only SkyMote nodes [3], which uses a CC2420 transceiver for wireless communication.

A. Simulation setup

Table I summarizes the most important simulation parameters. For the rest, we use the default values of Contiki parameters. We analyze the performance of several networks changing the total number of nodes in the network but always following the same idea: one main sender, one destination and some nodes in the middle whose only mission is to retransmit the WuS. We also vary the collision probability in the medium model separately for each radio type. In all cases, we perform

6 simulation runs for each particular scenario and we cut off the first and last 10 s of the duration of each simulation.

We implement 3 main MAC protocols for the analysis:

1) *ContikiMAC*: We use ContikiMAC [13] with the default parameters and turn on the phase optimization mechanism. For scenarios with more than 2 nodes, the nodes that are placed between the sender and the destination are only overhearing and do not participate actively in the communication.

2) *Single WuS*: In this protocol, only the WuS REQ is sent to wake-up the main destination. The main data is sent after the sync delay described in Section II. We implement 3 versions of this protocol: no sync delay (data transmission as soon as possible), medium sync delay and long sync delay. In the long case, the delay is long enough so that the WuS REQ can hop up to 9 times and arrive successfully at the main destination before the main data is sent.

3) *WuSACK*: Here when the main destination receives a WuS REQ, it transmits a WuS ACK back to the main sender. The sender will not transmit the main data until it receives the WuS ACK back.

For all of them, we use the Contiki implementation of CSMA on top of them without backoff exponential increments. RIME is used as the network layer for the main node in all scenarios because it is a lightweight protocol that let us focus on the MAC layer.

The currents and voltage values for the Sky platform are the nominal values obtained from the "Typical Operating Conditions" table of the datasheet [3]. On the other hand, the values for the WuR are borrowed from the experimental validation in [11]: 1.8 V of power supply, 8 mA current when transmitting the WuS, around 2 mA when actively receiving a WuS, and 1.944 μ W power consumption when idle listening.

To measure the power consumption of the main radio and main MCU, we use a combination of Powertrace and PowerTracker. For the power consumption of the WuR in TX and RX modes, we use the WurPowerTracker plugin provided in [11]. The idle listening power consumption of this secondary radio is not measured directly as it is always-on, so it is a constant added manually to our post-processing.

IV. RESULTS

The results presented in this section are an average of the overall data collected on the set of simulations. The 95% confidence interval indicates the reliability of our measurements.

A. Power consumption

1) *Ideal mediums*: The power consumption results in the ideal medium are shown in Fig. 6. The network radio mean power consumption only takes into account transmission, reception and idle listening modes for power consumption calculation, i.e., it does not consider the CPU and Low Power Mode (LPM). The values shown in the figure are the total sum of all nodes in each scenario.

The only scenario that is plotted for ContikiMAC is the 2-nodes network, which is extrapolated horizontally, because all the other scenarios running this protocol always consume

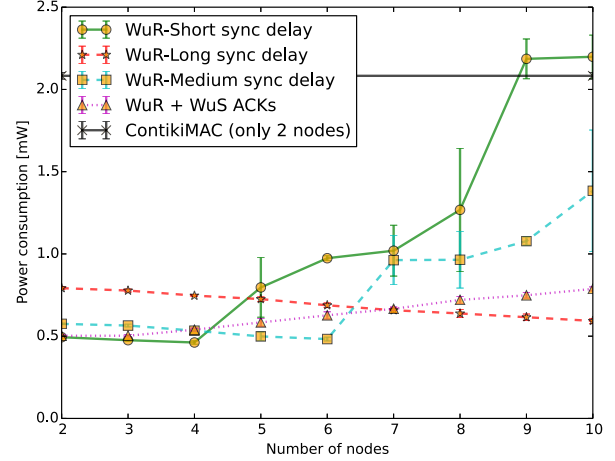


Fig. 6. Network radio mean power consumption. Ideal radio medium (no collisions).

more energy than its WuR counterpart. This is due to the large amount of idle listening and overhearing that is spent on the main radio in duty-cycled protocols like ContikiMAC. This is avoided with the WuR because the idle listening is done with the ultra-low power radio. In the WuR protocols, there is no overhearing of the main radio because if the node is not the main destination, then its main radio stays off.

Among the WuR protocols, we can divide the scenarios into 3 types depending on the network size (number of nodes): small, medium and large. In small networks, we see that WuR-Short sync delay is the protocol that performs the best in terms of power consumption, while the power consumption of protocols with intentional sync delay increases. This is because in a small network the main destination is near, so it receives the WuS and wakes up quickly. Then, the sync delay makes the destination wait in idle listening, wasting energy. As the number of nodes goes up, the WuS path length increases, but the sync delay at the transmitter stays the same. Hence, the idle listening wastage is reduced as the network size increases. This explains why the power consumption of WuR-Long delay decreases whereas the network size increases. WuR + WuS ACKs solves this problem because the sender gets to know in real time when the main destination is awake so it can send the main data.

WuR-Short sync delay deserves special attention at a medium network size of 5 nodes, where we see a sudden jump in the power consumption. As we increase the network size, there is a point in which the WuS path is so long that by the time the sender transmits the main data, the main destination has not received the WuS yet, so it is not awake and therefore is not ready to receive it. Consequently, the sender needs to retransmit the main data, wasting energy in useless early transmissions. When that happens, the transmitter starts over from the very beginning, sending the WuS and later on retransmitting the main data.

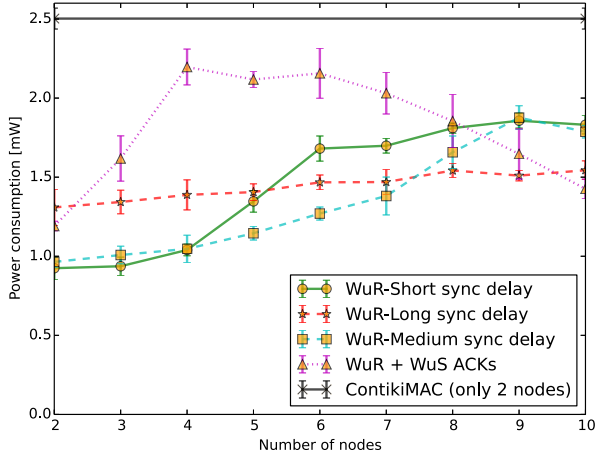


Fig. 7. Total radio mean power consumption. 20% probability of having collisions.

For WuR-Medium sync delay at that network size, this problem is avoided by the addition of the sync delay, which makes it the best protocol in terms of power consumption for medium network size. As a matter of fact, we can see that if we keep increasing the number of nodes, the problem arises at some point. In this case, this is at 7 nodes. On the other hand, the long sync delay is long enough so that the problem does not arise even in a 10-nodes network. Finally, in the WuS ACKs counterpart, this problem is not present for the same reasons as in smaller networks.

In large networks, there is another special point in the WuR-Short sync delay solution, at the 9-nodes scenario, where we find a new jump in the power consumption. In this case, the problem is that the WuS path is quite long. The second sequence of WuS REQ arrives at the main destination after the main data and ACK have been exchanged, so the destination wakes up again and waits for a new data packet that will never arrive because the sender has already sent it successfully. This increases the amount of idle listening on the main destination resulting in a higher power consumption.

In large networks, WuR-Long sync delay outperforms all other protocols because its long sync delay prevents the sender from retransmitting the main data. Finally, WuR + WuS ACKs consumes more energy than WuR-Long sync delay because it needs more WuR packets to be sent (for WuS ACKs), increasing a little bit the power consumption in the WuR, while the power consumption of the main radio remains the same.

2) *Collisions*: We run the same scenarios by adding a probability of collisions of 20% in each radio medium. The results are shown in Fig. 7. The overall power consumption is higher because of the retransmissions of the main data, but the behavior, in general, stays the same.

There is one particular change in behavior in WuS + WuS ACKs, where we can see that for small networks, there is a very high power consumption that increases proportionally

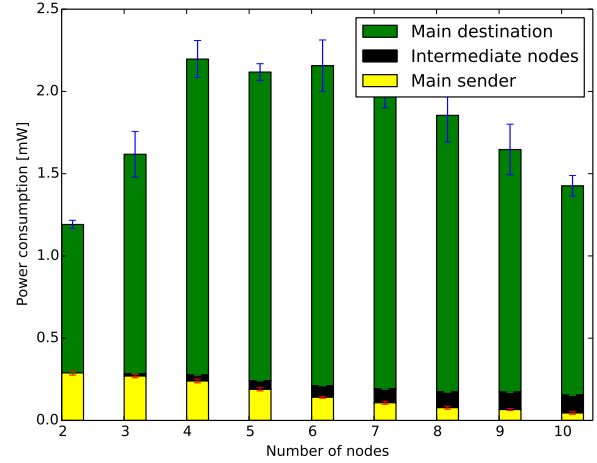


Fig. 8. Power consumption per type of node. Protocol WuR + WuS ACKs. 20% probability of having collisions.

with the network size. This is because most of the times the WuS sequence is successful, so the main destination wakes up. However, later on, some WuS ACK packet fails so the main sender does not transmit the main data. Meanwhile, the destination keeps listening on the main radio in vain. In consequence, the energy consumption increases due to idle listening on the main radio at the main destination. This is also reflected in Fig. 8, where we see clearly that the total power consumption is commanded mainly by the destination node. Whereas when the network is larger, there are several nodes in the middle and it becomes very hard to get all the WuS sequence successfully to wake-up the final destination, so there is less listening wastage. In addition, it is even more difficult to get the full WuS + WuS ACK sequence successfully to get to send a main data packet. So in the end, there is less listening on the destination and less number of main data packets transmitted by the sender. This behavior is because main radio packets are the ones that impact more on the overall power consumption while WuR packets retransmissions are insignificant.

3) *2-nodes ContikiMAC comparison*: We turn our attention now to figure 9, where we do consider the CPU and LPM to compare the total power consumption of the network. It is worth stressing out that when we compare the WuR protocols with the initial ContikiMAC scenario with only 2 nodes (horizontal line in the middle), we find that there is a threshold in the network size that determines whether the WuR outperforms ContikiMAC or not. This threshold turns out to be in a network size of between 5 and 6 nodes. So for example, this means that having 2 nodes communicating with ContikiMAC is better in terms of power consumption than having 7 nodes communicating with a WuR protocol. In contrast, 2 nodes with ContikiMAC is worse than 4 nodes with a WuR protocol. This threshold barely changes for most protocols when we increase the probability of collisions. WuR

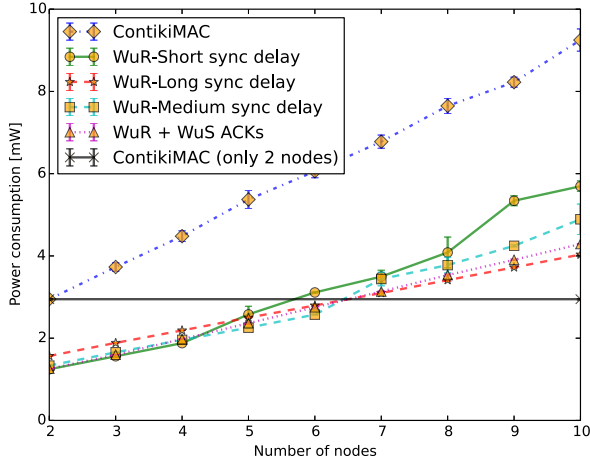


Fig. 9. Network mean power consumption. Ideal radio medium (no collisions).

+ WuS ACKs is the only one that presents a large variation of this threshold, because of the high power consumption for small networks that we described previously.

B. Latency

The latency behavior is led by the number of main data retransmissions. This is why the shape of the lines for each protocol in Fig. 10 is similar to that of the power consumption. So again, WuR-Short sync delay is the best one for small network sizes, WuR-Medium delay is so for medium networks and WuR-Large delay for large networks. In contrast, when there is a chance of collisions in the medium, it is always better to provide the minimum amount of intentional delay to the protocol, as shown in Fig. 11. This is because each retransmission adds this delay to the total latency, so if there are going to be several retransmissions, it is better to add the minimum delay possible for each one. It is noticeable that WuR + WuS ACKs provides the worst performance in terms of latency because it is the one that requires more number of retransmissions for each successful communication.

ContikiMAC latency is not plotted because it is very high and the figure is easier to read without it. Its value is around 100 ms for the ideal medium and 120 ms for the collision scenario (20% for both mediums). This is a reasonable number, considering that we used the default channel check rate of ContikiMAC which is 8 Hz. This corresponds to a period of 125 ms.

C. Packet Delivery Ratio (PDR)

Regarding the reliability of the protocol, we see in Fig. 12 that the WuS sequence length strongly affects the PDR. This is because it is more difficult to get several packets in a row successfully than just a few of them. The PDR decreases with the increase of the network size, but also with the addition of WuS ACKs, which doubles the amount of WuS packets for each scenario reducing significantly the reliability

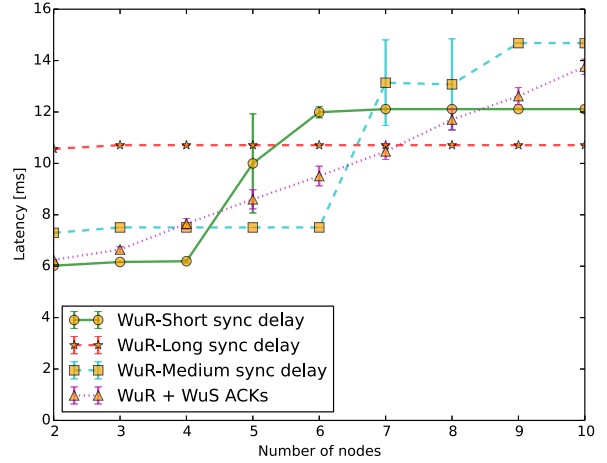


Fig. 10. End-to-end latency. Ideal radio medium (no collisions).

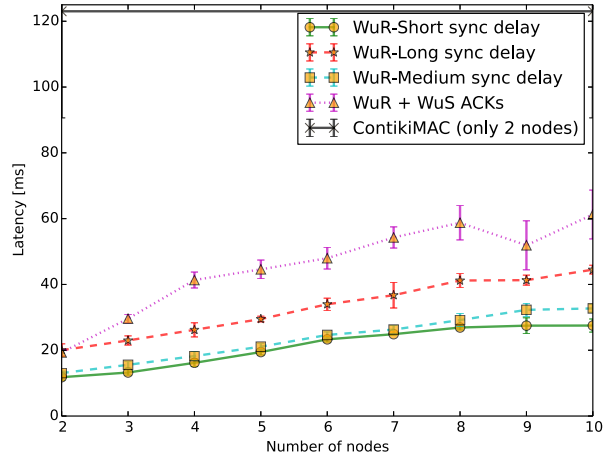


Fig. 11. End-to-end latency. 20% probability of having collisions.

of the communication. In consequence, the PDR is tightly controlled by the RX success probability of the WuR medium. In addition, we present the resulting PDR when the probability of collisions for the WuR medium is decreased to 10%, while the corresponding probability for the MR medium is kept at 20%, in Fig. 13. As we can see the PDR significantly increases for all WuR solutions, emphasizes that the reliability of the WuR medium is of crucial importance.

The PDR for the ideal medium is not shown because it is always 100%. In fact, even if the MR medium presents collisions, but the WuR is ideal, then the PDR is still 100% thanks to the retransmissions allowed by CSMA. This reinforces the conclusion that the PDR is mandated by the number of collisions in the WuR medium.

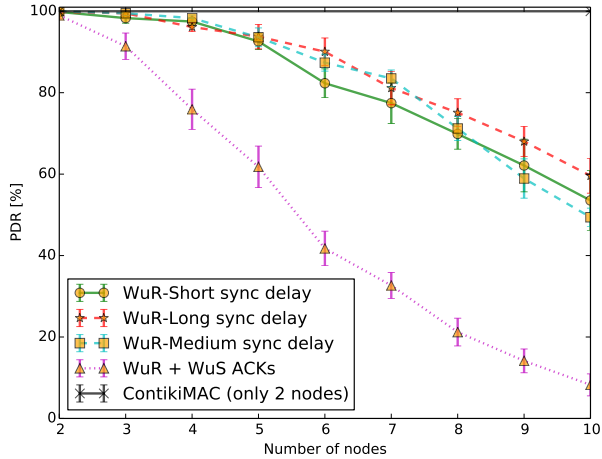


Fig. 12. Packet Delivery Ratio. 20% probability of having collisions.

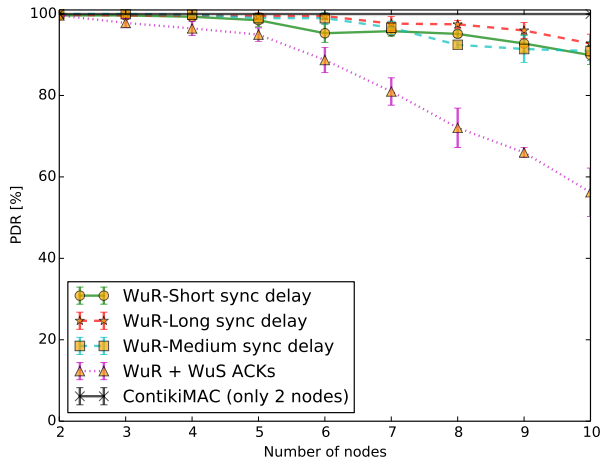


Fig. 13. Packet Delivery Ratio. 20% probability of having collisions in the main radio and 10% in the WuR

D. Simulation conclusions

In the light of the obtained results, some conclusions can be drawn. First, WuR is not always the final solution to the latency-energy tradeoff in multi-hop WSN. As expected, WuR achieves better power consumption and latency than traditional duty-cycled protocols, in particular, ContikiMAC, considering an ideal medium with the destination directly in the WuR range of the sender. However, due to the range mismatch between MR and WuR, WuS packets would be relayed by intermediate nodes to reach the final destination. In that case, WuR still achieves better performance if the relaying nodes involved in the communications are no more than 5, as shown in Fig. 9. For large networks (for 6 or more relaying nodes), the latency obtained with WuR is still lower than for ContikiMAC, but the overall power consumption is higher. Collisions on the WuR medium also have a very significant impact on

overall performance, especially on the achieved PDR. It is of crucial importance that WuS packets reach the destination or WuR presents severe underachievements. We are considering mechanisms such as an exponential backoff on WuS packets to mitigate this. Finally, we advocate to not use acknowledgments for WuS packet. Not only the power consumption is slightly increased, but the PDR is significantly reduced together with an increase in the latency. Obviously, WuR-Short, WuR-Medium and WuR-Large sync delays respectively match small, medium and large networks. We will investigate how we can dynamically adapt this delay regarding the network size.

V. RELATED WORK

A complete survey of the WuR technology was published in [2]. We conclude from that article that most of the work done so far in WuR is mainly focused on the hardware side and not too much attention has been paid to the networking counterpart. Even the protocols surveyed in that paper are in general very specific to a particular prototype or application ([14], [15]).

An interesting comparative analysis between preamble sampled MAC protocols and WuR receivers in WSN has been proposed in [4]. There, it is shown that in practice, the range mismatch requires a denser deployment of the WSN, or the need to transmit the WuS at a high output power. However, there is no further analysis of the details of the communication protocol when the node density is increased. This is likely to increase the competition in the WuR medium, leading to more collisions. As we have seen in Section IV, the reliability of the WuR medium significantly impacts on the performance of WuR.

Simulations in OMNeT++ [8] have been done in [5] in order to prove that WuR constantly allows for substantial energy savings, higher PDR, lower latency and less complicated software implementations. In spite of the fact that the authors provide several variations of the network type and details on how to reproduce the analysis, the comparison considers neither the relay of the WuS to reach a destination nor acknowledging WuS packets.

T-ROME [16] is one of the few solutions that propose to use intermediate nodes to retransmit the WuS between the source and the destination. The reactive protocol operates dynamically choosing the most appropriate stopover nodes in case the sink is not reachable within one communication hop. Besides, it allows sending several main data packets in a row once a communication link is established. However, the authors do not analyze how large can the network be or whether it is actually interesting to have WuS ACKs. We showed that there is a clear threshold in the network size for WuR to achieve the best performance.

VI. CONCLUSIONS AND FUTURE WORK

In this article, we investigated how Wake-Up radio can increase the network performance of multi-hop WSN. For this, we analyzed the power consumption, latency, and PDR of a traditional duty-cycled MAC protocol (ContikiMAC) and

4 variations of the WuR protocol. In Section IV, we find that WuR protocols always outperform traditional duty-cycled MAC protocols for the same number of nodes. Although if we compare a scenario with only 2 nodes using duty-cycle, then there is a threshold in the number of nodes that we can have in the middle relaying the WuS for it to be outperforming.

We show in Section IV that the presence of collisions significantly affects the network performance, wasting energy and degrading the PDR and the latency. In fact, WuS ACKs is only worth it if the WuR medium is free of collisions. Using exponential backoff in CSMA can mitigate this issue on the WuR at the cost of increasing the latency.

A protocol that adapts dynamically the sync delay, according to the network size, can achieve the minimum power consumption. The drawback is that the latency would be worst in real scenarios where we have collisions. Our results show that, in the collisions case, it is better to have the minimum sync delay possible no matter the network size. So here we show that there is still a tradeoff between power consumption and latency in WuR.

Medium delay protocol appears to be a reasonable compromise between latency and power consumption for collision scenarios.

Based on the results of this work, we understand that the WuR packet retransmissions have a minor impact on the power consumption and that the PDR is heavily affected by the WuR medium reliability. Accordingly, we could just increment the maximum number of retransmissions for the WuR, while keeping the same parameters in the main radio. This way we would have more chances to deliver the WuR packets, countering the medium reliability and with a minor impact on power consumption. For this to happen, we must add a new mechanism for each relay node to find out if the next relay in the path has received the packet. This can be easily implemented overhearing the WuR medium. This will also bring a tradeoff between PDR and latency, as the increased number of retransmissions comes by the hand of higher latency, as we saw in the results with collisions.

As for the power consumption, we show that it is not the retransmission of WuR packets what impacts more, but the retransmission and idle listening on the main radio. In conclusion, it does not seem to be necessary to duty-cycle the WuR, as is suggested in some works [2].

In the future, we would like to extend our work with experiments on hardware prototypes. Furthermore, we leave an open question about the design of the routing protocol for WuR systems with the range mismatch issue. The design of such protocol is out of the scope of this article and is a future challenge. However, the always-on nature of the WuR can provide information on the channel characteristics. We believe that the future point-to-point routing protocol can self-adapt its parameters taking advantage of that.

The conclusions of this paper, together with those of the related work in the field, drive the advancement in WuR into specific applications where all the conditions described must

be met in order for it to be a successful improvement over traditional duty-cycled MAC protocols.

ACKNOWLEDGMENTS

This work is part of the project *WakeUp* funded by the French National Research Agency (ANR).

REFERENCES

- [1] F. Alfayez, M. Hammoudeh, and A. Abuarqoub, "A survey on mac protocols for duty-cycled wireless sensor networks," *Procedia Computer Science*, vol. 73, pp. 482 – 489, 2015, international Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015).
- [2] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2117–2157, Fourthquarter 2017.
- [3] "T-mote sky datasheet." [Online]. Available: <http://www.eecs.harvard.edu/konrad/projects/shimmer/references/tmot-sky-datasheet.pdf>
- [4] R. Su, T. Watteyne, and K. S. J. Pister, "Comparison between preamble sampling and wake-up receivers in wireless sensor networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec 2010, pp. 1–5.
- [5] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 674–687, April 2016.
- [6] M. Magno, V. Jelcic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, "Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 633–644, April 2016.
- [7] F. Ait Aoudia, M. Gautier, and O. Berder, "OPWUM: Opportunistic MAC Protocol Leveraging Wake-Up Receivers in WSNs," *Journal of Sensors*, Jan. 2016. [Online]. Available: <https://hal.inria.fr/hal-01244800>
- [8] A. Varga, *OMNeT++*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–59.
- [9] "The network simulator - ns-3," 2009. [Online]. Available: <https://www.nsnam.org/overview/what-is-ns-3/>
- [10] D. Benedetti, C. Petrioli, and D. Spenza, "GreenCastalia: An Energy-Harvesting-Enabled Framework for the Castalia Simulator," in *Proceedings of ACM ENSSys 2013*. Rome, Italy: ACM, Nov 14 2013, pp. 7:1–7:6.
- [11] R. Piyare, T. Istomin, and A. L. Murphy, "Waco: A wake-up radio cooja extension for simulating ultra low power radios," in *EWSN*, 2017.
- [12] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, ser. SenSys '07. New York, NY, USA: ACM, 2007, pp. 335–349.
- [13] A. Dunkels, "The contikimac radio duty cycling protocol," 05 2012.
- [14] F. Dressler, M. Mutschlechner, B. Li, R. Kapitza, S. Ripberger, C. Eibel, B. Herzog, T. Hönig, and W. Schröder-Preikschat, "Monitoring bats in the wild: On using erasure codes for energy-efficient wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 12, no. 1, pp. 7:1–7:29, Feb. 2016.
- [15] N. Simon, J. Bordoy, F. Hfingler, J. Wendeborg, M. Schink, R. Tannhuser, L. Reindl, and C. Schindelbauer, "Indoor localization system for emergency responders with ultra low-power radio landmarks," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2015, pp. 309–314.
- [16] T. Kumberg, M. Schink, L. Reindl, and C. Schindelbauer, "T-rome: A simple and energy efficient tree routing protocol for low-power wake-up receivers," 02 2017.