The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN

Nour El Hoda Djidi, Sebastian Sampayo, Julien Montavont, Antoine Courtay, Matthieu Gautier, Olivier Berder, Thomas Noël

▶ To cite this version:

Nour El Hoda Djidi, Sebastian Sampayo, Julien Montavont, Antoine Courtay, Matthieu Gautier, et al.. The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN. IEEE Wireless Communications and Networking Conference (WCNC), Apr 2022, Austin, United States. hal-03648488

HAL Id: hal-03648488
https://hal.inria.fr/hal-03648488

Submitted on 21 Apr 2022

The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN

Nour El Hoda Djidi *, Sebastian Sampayo † Julien Montavont †, Antoine Courtay *,
Matthieu Gautier *, Olivier Berder *, Thomas Noël †
* Univ Rennes, CNRS, IRISA, Lannion, France
Email: {nour-el-hoda.djidi,antoine.courtay,matthieu.gautier,olivier.berder}@irisa.fr
† ICube Laboratory, CNRS, University of Strasbourg, France
Email: {sampayo,montavont,noel}@unistra.fr

***Abstract*—Synchronized MAC protocols are now considered as the ultimate solution to access the medium in wireless sensor networks. They guarantee both high throughout and constant latency and achieve reasonable energy consumption performance. However, synchronization is achieved at the cost of a complex framework with low flexibility on its parameters that is not suitable for some network topologies or application requirements. By contrast, asynchronous MAC protocols are versatile by nature but suffer from the tradeoff between energy consumption and latency. However, the addition of Wake-up Radio (WuR) can reduce the energy consumption of such protocols while maintaining very low latency thanks to its always-on feature and ultra-low power consumption. In this article, we present WuR-based Multi-hop Multi-channel (W2M), an asynchronous MAC protocol for wireless sensor networks. We also provide a fair comparison with Time Synchronized Channel Hopping (TSCH) through an extensive simulation campaign based on Contiki-NG and Cooja. Our results show that in low traffic scenarios, W2M outperforms TSCH in reducing both the energy consumption and the latency (at least 68% of energy is saved), but at the cost of slightly lower reliability.**

***Index Terms*—Wake-up radio, TSCH, MAC protocols, Wireless Sensor Networks**

I. INTRODUCTION

Thanks to the dramatic decrease in the cost of sensors and other electronic components, wireless sensor networks have been widely deployed in the recent years to monitor various areas, giving birth to many applications and making everything smarter (smart cities, smart factories, smart agriculture, smart homes...). Energy consumption still represents the major drag on an even more massive invasion of sensors, since the more nodes you deploy, the less often you want to have to change the batteries. As radio is generally the main source of energy consumption, many efforts were devoted to limit the time transceivers are on through duty-cycling, either with synchronous or asynchronous MAC protocols. Both approaches were often compared and each of them has proved its superiority, depending on networks topologies and application requirements. Nevertheless, collisions, idle listening and overhearing make asynchronous protocols generally less energy-efficient and reliable than synchronous protocols.

On the other hand, Wake-up Radio (WuR) emergence could change (again) how nodes access the wireless medium. The very simple architecture of WuR makes it possible to keep it always listening while consuming quasi nothing (typically few microwatts but sometimes even less [1]), which enables pure asynchronous communications with negligible latency while mitigating idle listening and overhearing. WuR-based MAC protocols were proved to be very efficient for many different network topologies when compared to classical asynchronous protocols [2]. WuR is however a matter of trade-off since to decrease as much the energy consumption comes at the price of a reduced radio range.

In this article, we introduce WuR-Based Multi-hop Multi-channel (W2M), an asynchronous MAC protocol for wireless sensor networks leveraging WuR. This protocol overcomes the WuR short radio range by introducing relays and mitigates collisions thanks to multi-channel communications. We also aim at determining if a WuR-based asynchronous protocol can have significant advantages over synchronous protocols and investigate the situations where WuR outperforms. To obtain the fairest comparison, Time Synchronized Channel Hopping (TSCH), considered as one of the most efficient synchronous MAC protocols [3], was chosen for simulation in different scenarios. To the best of our knowledge, we are the first to provide such an analysis. The contributions of the present article are threefold: 1) a novel WuR-based protocol that enables Multi-hop and Multi-channel communications, namely W2M; 2) an implementation of this protocol in Contiki-NG; 3) an exhaustive comparison of W2M and TSCH for different traffic rates in terms of packet delivery ratio, latency and energy consumption.

The rest of the paper is organized as follows. Backgrounds on synchronous protocols, WuR, and WuR-based protocols are given in Section 2. Our WuR-based multi-channel protocol, W2M, is detailed in Section 3. Section 4 is devoted to the comparison methodology, while the simulation results are presented in Section 5. Some conclusions are finally drawn in Section 6.

II. BACKGROUND AND RELATED WORK

A. *Synchronized protocols*

Synchronized protocols require time synchronization (hence the name) between nodes to control how the wireless channel is accessed. Nodes agree on a common schedule that defines when a node can transmit, receive or sleep. However, the level of synchronization differs between protocols. For example, slotted schemes use Time Division Multiple Access (TDMA) [4] and therefore require a tight synchronization while protocols based on a common active/sleep period are more tolerant regarding time drift [3].

Time Slotted Channel Hopping (TSCH) was introduced in 2012 as a new MAC behavior for the IEEE 802.15.4 standard [5]. TSCH enables deterministic communications with guaranteed throughput and bounded delay. In addition, multi-channel and channel hopping increase network capacity and reliability. TSCH divides time into slots of fixed duration (referred to as timeslots) in which a node can transmit, receive or sleep. A slotframe is a set of timeslots that repeats over time. A TSCH link is an oriented communication between a source and a destination in a specific timeslot and channel offset. The latter specifies the effective channel in which the communication will occur. The collection of TSCH links forms the TSCH schedule that is shared between all network nodes. Centralized schedulers have been proposed [6] but require to first collect network statistics at a central station before broadcasting the resulting schedule. Such operations represent a pure protocol overhead. By contrast, distributed schedulers require that nodes negotiate with their neighbors to build a consistent schedule [7]. However, such solutions generally require a stable environment with low traffic conditions.

Other approaches such as Orchestra [8] enable nodes to autonomously compute a local schedule without signaling overhead or any central or distributed scheduler. Orchestra works with the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [9] to build the schedule. A node locks a timeslot regarding its identification number (ID). Once RPL converged, a node knows its RPL neighbors and therefore can derive their timeslots from their IDs. Timeslots locked by the nodes can be shared, receiver-based, or sender-based. Any node can use a shared timeslot to transmit or receive using a contention-based protocol.

An Orchestra schedule is composed of multiple slotframes, one per traffic type. A typical Orchestra schedule is composed of three slotframes: one for transmitting the IEEE 802.15.4 Enhanced Beacon (EB) using sender-based timeslots, one for the RPL signaling using a unique shared timeslot, and one for the application traffic using either receiver or sender-based timeslots. Fig. 1 illustrates the Orchestra schedule for a node whose ID is 2. Slotframe types and sizes are pre-configured on each node. The slotframes are organized by priority to resolve potential timeslot conflict.

B. *Wake-up Radio*

Thanks to Wake-up Radio (WuR), wireless communications are taking a disruptive step towards very low power consumption. This technology comes as a secondary ultra-low-power radio receiver that can continuously monitor the wireless channel while consuming orders of magnitude less power than traditional transceivers [2]. Fig. 2 depicts how WuR are mostly used. These devices wake up the node main radio or other sleeping subsystems using interrupts when a specific signal, called Wake-up Signal (WUS), is detected. Before transmitting a data packet, a Source (S) sends a WUS to wake up the main radio of the Destination (D). WUS could be either sent by the WuR or by the main radio. Nodes should therefore deal with two communication channels: the WuR radio channel (for WUS transmission) and the main radio channel (used for effective communication).

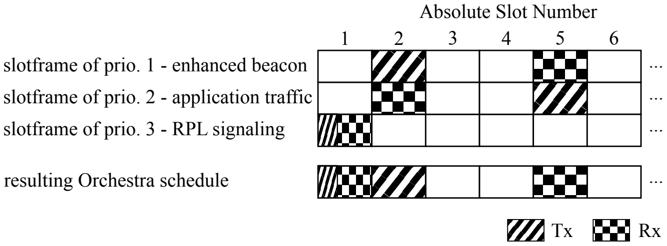


Fig. 1: Orchestra schedule example for a node of ID 2.

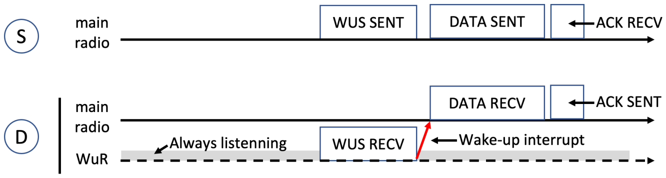


Fig. 2: Wake-up mechanism using WuR.

One of the main benefits of WuR is to enable *pure* asynchronous communication that can significantly increase the energy efficiency of communications and reduce the latency [10]. To ensure ultra-low power consumption, state-of-the-art WuRs are usually characterized by their lower sensitivity and their lower bit rate compared with traditional IoT transceivers such as IEEE 802.15.4 [1]. While most prototypes rely on very simple modulation schemes, typically On-Off Keying (OOK), interesting approaches were proposed first to enhance the filtering process of parasitic signals, then to address existing standards, e.g. 5G IoT networks [11]. A low sensitivity induces a short radio range of few tens of meters [12] for state-of-the-art WuRs. To tackle this drawback, WUS can be sent at a higher transmission power (which in turns increases the energy consumption), or dedicated MAC protocols using relaying techniques can be used [13].

WuR-based MAC protocols have been compared to asynchronous MAC protocols [10], [14]. Those studies show that they always outperform traditional duty-cycled solutions. However, how they compete with synchronous MAC protocols remains an open question that we address in the next sections.

III. WuR-based Multi-hop Multi-channel protocol

A. *Protocol concept*

This section presents W2M, our novel asynchronous MAC protocol based on WuR. In this protocol, nodes can communicate via two communication channels. The WuR channel is only used for signaling while the main channel is used to exchange control and data packets. When a source has DATA to send, it first sends a Wake-Up Signal (WUS) through the WuR channel and waits for a delay (SYNC_DELAY) to turn on its main radio. This delay corresponds to the time needed for receiving the WUS by the destination, and it depends on the number of hops and the WUS duration. Upon reception (all nodes continuously listen on this channel thanks to the WUR ultra-low power consumption), the destination turns on its main radio and broadcasts a Ready To Receive (RTR) message over the main channel, that triggers the DATA transmission from the source. Then, the destination sends back an Acknowledgement (ACK). In case of errors (no ACK or no RTR), the source restarts the process from the WUS transmission. To limit collisions, CSMA is used with a random backoff, and the number of packet re-transmissions is limited.

Because of the low sensibility of WuR, WUS are not necessarily received by nodes directly reachable through the main radio. To deal with this range mismatch, specific nodes are introduced as relays between the source and the destination that only serve to relay the WUS. However, extending the range of WuR with such techniques is limited as using more than five relays could degrade the protocol performance [14].

To increase the network capacity, W2M uses multiple channels to exchange messages over the main radio. Before transmitting a WUS, a source randomly selects the operating channel among the sixteen IEEE 802.15.4 channels at the 2.4 GHz. As a result, each WUS includes the destination address and the operating channel of the pending communication, plus the next relay address. Note that the WuR only uses a single radio channel. The principle of this MAC protocol is shown in Fig. 3.

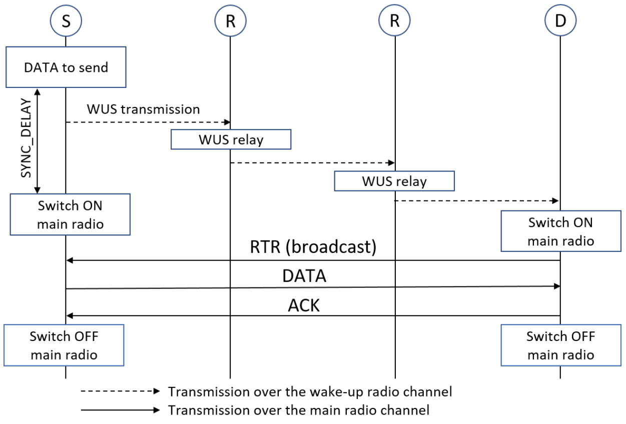
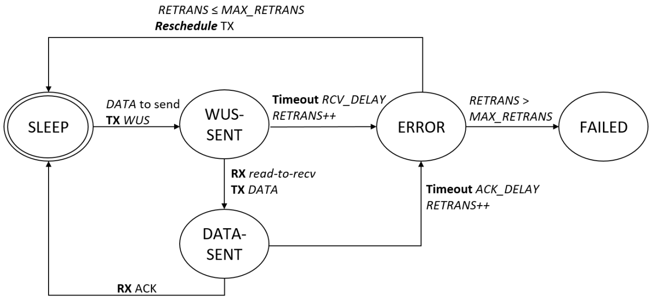
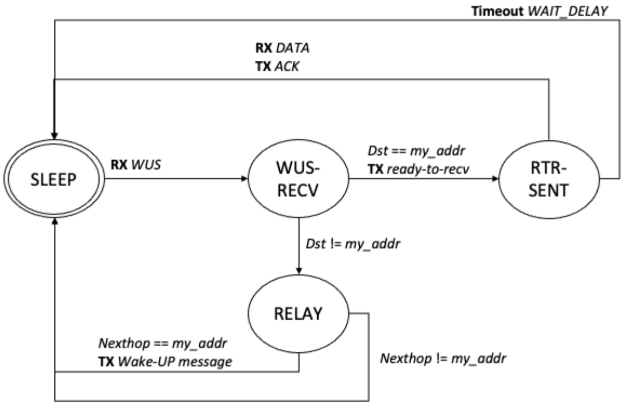


Fig. 3: Principle of W2M protocol.



(a) Transmit mode.



(b) Receive mode.

Fig. 4: Statesharts of the protocol implemented in all nodes.

B. *Implementation*

The details of the implemented W2M protocol are shown through the statecharts in Fig. 4 for both transmit and receive modes. In the transmit mode (Fig.4(a)), the node that has DATA to send, passes from sleep state to WUS-SENT state and transmits the WUS. Then, it waits during SYNC_DELAY before turning on the main radio to receive RTR. If no RTR is received after RCV_DELAY, the node comes back to sleep and reschedules the transmission if the number of retransmitted packets does not exceed MAX_RETRANS, or drops the packet otherwise. If the node receives RTR, it sends DATA and waits for ACK. If no ACK is received after ACK_DELAY, the node reschedules the transmission if the number of retransmissions does not exceed MAX_RETRANS and comes back to the first state. If ACK is received, then the node goes back to sleep state.

In the receive mode (Fig.4(b)), when the node receives the WUS, it passes through the state WUS-RECV and checks the destination address contained in the WUS. If it matches its address, then the node tunes its channel number to the one announced in the WUS, sends RTR, and waits for DATA. If no DATA is received after WAIT_DELAY, it goes to the sleep state or sends ACK otherwise. If the node is not the destination, then it checks if the next-hop address matches its address meaning that it should act as a relay. In this case, it updates the next-hop address and forwards the WUS towards the destination before going back to the sleep state. If the node is not the destination nor the next-hop, it simply drops the WUS and goes back to the sleep state.

IV. SIMULATION SETUP

W2M has been implemented into Contiki-NG which is one of the most popular operating systems for resource-constrained devices. We based our development on Waco [15], a Contiki extension that provides support for WuR. Furthermore, we ported Waco to Contiki-NG as it was originally developed for Contiki-2. The network is simulated by the Contiki network simulator COOJA. We choose this framework over other solutions to ease future experimentations, as the used software for simulations could be directly exported to real sensor boards. In the following, we compare W2M to TSCH coupled with Orchestra natively supported by Contiki-NG.

The simulated network is composed of 30 nodes organized as a grid. The number of neighbors of each node ranges from 2 to 4 regarding its position in the grid (only nodes located at cardinal points are neighbors). Fig. 5 illustrates this topology. Node 1 operates as the sink, being the final destination of data traffic. The other nodes periodically generate a data packet and forward it to their next hops in the grid. As we focus on the MAC layer performance, next hops are calculated offline to limit interference from a complex and high overhead routing protocol. They are pre-selected by minimizing the number of hops to reach the sink. In the case of equidistant next hops (e.g. Node 8 can reach the sink through Node 2 or Node 7), one node is randomly selected. The resulting routes are illustrated by dashed arrows in Fig. 5. A similar routing structure can be obtained with RPL [9]. Considering W2M, two intermediate relay nodes are introduced between each neighbor to deal with the range mismatch between the main radio and the WuR. For example, if Node 3 wants to transmit a data packet to Node 2, it first sends a WUS destined to Node 2 that is relayed twice by two additional nodes before reaching Node 2.

Orchestra is configured with two slotframes: one of 397 timeslots (default) for EB transmission and one of 31 timeslots for data transmission. The first is of higher priority of the latter in case of collision. The size of each slotframe allows to dedicate one timeslot to each node and therefore enables sender-based dedicated timeslots resulting in collision-free communications. For the W2M, WUS contains 16 bits: the first 6 bits are for the destination address, the next 6 bits are for the next relay address, and the last 4 bits are for the channel number. As the range of the WuR is short, nodes that are far from each other can use the same address, so 6 bits addresses can cover all the node addresses of even a large network. Table I gives all simulation parameters.

V. RESULTS

The results presented in this section are collected and averaged from more than 133 h of simulated time, corresponding to more than 650k generated data packets. The standard deviations showed in the figures indicate the precision of our results. At each run, the TSCH nodes require a warm-up period of 13 mins to discover their time sources. This period is removed from the statistics. High traffic of 1pkt/10s and low traffic of 1pkt/2mins are considered.

Parameter	Value
Radio range ratio (Main/WuR)	3
Datarate Main/WuR	250 kbps / 10 kbps
Medium model	Unit Disk Graph Model
Traffic type	Convergecast
Payload size	60 B
Packet generation period	5 s / 120 s
Number of packets per node	500 / 42
Wake-up Beacon/ACK/RTR size	2 B / 3 B / 1 B
MAX_RETRANS	7
CSMA-Min-BE/Max-BE	3 / 5
CSMA-Max-Backoff	5
SYNC_DELAY / RCV_DELAY	3.2 ms / 16 ms
ACK_DELAY / WAIT_DELAY	2.4 ms / 9.6 ms
TSCH timeslot duration	10 ms
EB transmission period	16 s
Orchestra slotframe size EB/DATA	397 / 31
Channel hopping sequence	15, 25, 26, 20
Orchestra slot type	Sender-based dedicated
Main node	Zoleria Z1
Main node supply voltage	3.3 V
Main node TX/RX	17.4 mA / 18.8 mA
WuR prototype	[1]
WuR supply voltage	3.3 V
WuR TX/RX/Idle current	17.4 mA / 80 μA / 7.6 μA

TABLE I: Simulation parameters.

The packet delivery ratio is presented in Fig. 6. Not a single packet is lost with TSCH in the high traffic scenario, thanks to dedicated timeslots. However, nodes located farthest away from the sink lost a few packets in the low traffic scenario (e.g. Node 18 or Node 30). On occasion, those nodes become out of synchronization and transmit while their next hops do not listen. The low traffic rate prevents periodic clock synchronization, and the drift is not always compensated by the guard time. By contrast, W2M faces more packet loss (approximately 2%) in both traffic types compared to TSCH. Collisions on WUS mainly contribute to packet loss because they are transmitted over a single channel and at a low datarate.

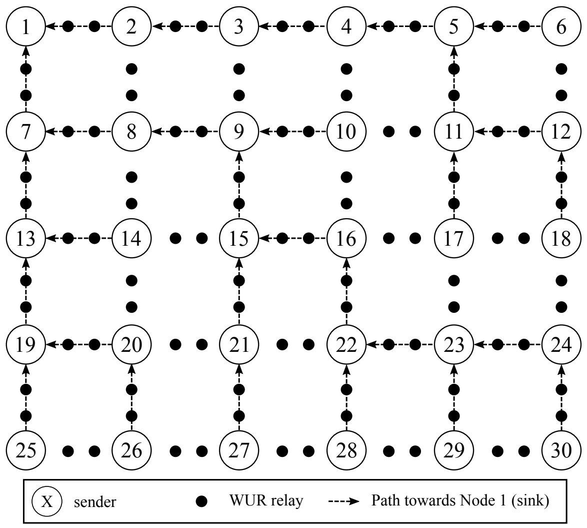


Fig. 5: Network topology and routing structure.

In addition, at least 3 WUS are transmitted per communication over the main radio due to relays, increasing the chance of collisions. However, once the wake-up procedure is successful, risks of collision are drastically reduced thanks to multi-channel communications on the main radio.

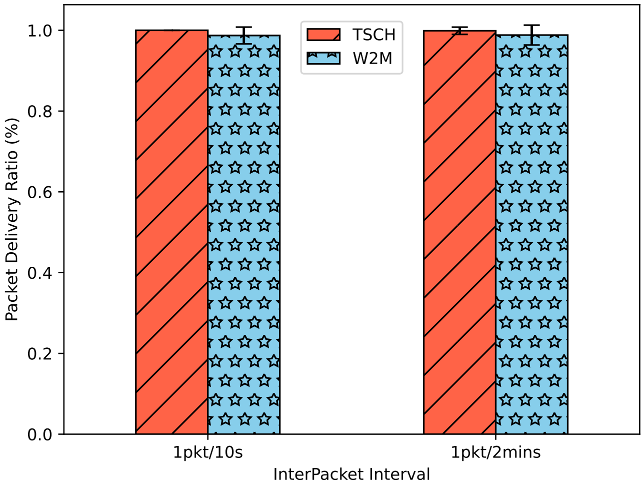


Fig. 6: Packet Delivery Ratio.

Fig. 7 presents the end-to-end delay for the data traffic. Delays are calculated from the generation of the data packet at the application layer of the source until it reaches the application layer of the sink (i.e. Node 1). Obviously, the more the number of hops between a source and the sink is, the higher the delay becomes. As we can see, W2M presents lower and more stable delays than TSCH. Being purely asynchronous, W2M allows nodes to transmit at any time. The delay is only affected by the wake-up procedure (relaying WUS and broadcasting RTR) that accounts for less than a few ten milliseconds. Collisions explain the observed variation as a single collision increases the delay by approximately 20 ms at most. By contrast, TSCH presents larger delays (sometimes larger than 10 s while W2M is bounded at less than 2 s). In our configuration, a node can only send one data packet every 310 ms. If a node misses its timeslot, it would wait for up to 310 ms before its next transmission opportunity. This is the counterpart of collision-free communication with Orchestra in which a timeslot (at least) has to be dedicated to each node. In addition, the large variations observed for TSCH are related to limited transmission opportunities and collisions. Nodes that relay many packets (such as Node 7) are likely to queue some of them, but each queued packet takes a significant extra delay related to its rank in the queue. Finally, intra-node and inter-node collisions can still occur in TSCH. The former refers to a situation in which a collision occurs between the EB and the DATA slotframes, deferring the transmission of the data packet. An inter-node collision occurs when two neighbor nodes respectively transmit a data packet and an EB in the same timeslot on the same channel. Due to the collision, the data packet is scheduled for retransmission in the next slotframe, increasing the overall delay accordingly.

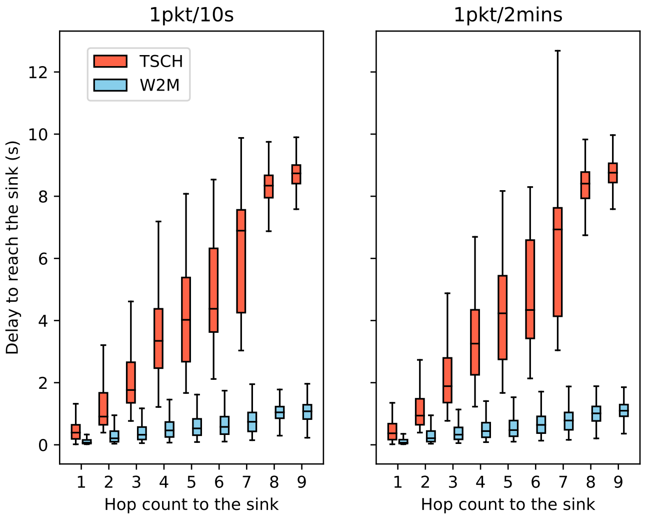


Fig. 7: End-to-end delay.

Finally, Fig. 8 presents the energy consumption breakdown. Nodes are grouped by the number of direct children they have in the routing path: leaf (no child, e.g. Node 6), one child (e.g. Node 3), and two children (the maximum, e.g. Node 7). MCU is the energy consumption of the microcontroller (MCU) of the main node that performs the active tasks of the node (managing sensors, computing algorithms, acquiring data, etc.). This is typically determined by the mean current consumption of the MCU in active mode. LPM is the Low Power Mode of the main node, i.e. when the aforementioned MCU is sleeping. TX and RX are respectively the energy spent by the main radio module (CC2420) in transmission and reception/idle. Finally, WuR_Tx is the energy spent by the WuR transmitting the WUS, WuR_Rx and WuR_MON are respectively the energy spent by the WuR when it is receiving a WUS and when it is only monitoring the channel.

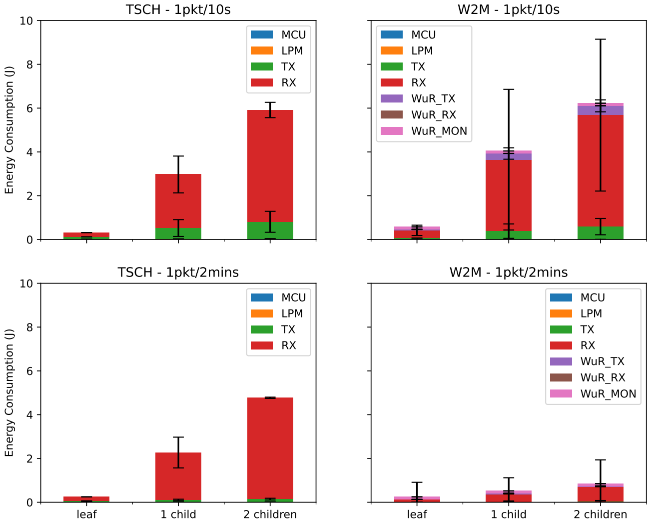


Fig. 8: Energy consumption.

Considering TSCH, the first factor of energy consumption is the number of direct children. As we can see, the nodes that consume the most are the ones with 2 direct children (e.g. Node 7) because such nodes may receive two data packets per DATA slotframe. Even if there is no effective transmission during those timeslots, they still switch on their radio and listen for a short amount of time (required to detect the frame preamble plus the guard time) before deducing that there is no packet on air. This is clearly shown in the 1pkt/2mins scenario, which presents almost the same energy consumption for RX than the 1pkt/10s scenario while 10 times less packets are exchanged. But the number of forwarded packets still has a slight impact on energy consumption. However, this factor impacts less energy consumption than the number of direct children. For example, Node 2 and Node 5 respectively count 8 and 5 nodes in their sub-graphs but Node 5 consumes 33% more than Node 2 because of its extra receiving timeslot in the DATA slotframe while it forwards 37% less packets. While the energy consumption mainly depends on the schedule in TSCH and remains almost constant in both traffic scenarios, it is directly related to the traffic in W2M. Although the consumption levels are similar between W2M and TSCH in the high traffic scenario, they are drastically reduced for W2M in the low traffic scenario (up to 78% for nodes with 2 children). In W2M, the nodes that consume the most are those that forward more packets. This explains the high variation in the energy consumption of nodes that have 1 child and 2 children because those sets include nodes with various sub-graph sizes. However, the overall energy budget of W2M could be slightly increased because of the relay nodes that are used to only relay the WUS. Nevertheless, those nodes consume between 82% and 90% less energy than nodes with 2 children thanks to the ultra-low power consumption of WuR.

VI. CONCLUSIONS

In this article, we proposed a novel asynchronous Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN and investigated its efficiency in front of the reference of synchronous MAC protocols ”TSCH”. This MAC protocol is implemented in Contiki-NG and is compared with TSCH in different traffic scenarios. We use Orchestra to define the TSCH schedule as it does not generate any protocol overhead. The results presented in Section V show that W2M outperforms TSCH in terms of end-to-end delay and energy consumption at the cost of fewer guarantees on reliability. It is sweet revenge for asynchronous protocols as they are more and more abandoned in favor of synchronous protocols. TSCH schedules such as those proposed by Orchestra are too rigid to adapt to the traffic and can not scale without increasing significantly observed delays. Global coordination remains essential to define a more optimal schedule but at the cost of a large signaling overhead. Being purely asynchronous, W2M is flexible and fully distributed by nature. Thanks to WuR, idle listening and overhearing (the two main drawbacks of asynchronous protocols) are solved, achieving performance close or even better than TSCH.

Encouraged by the results presented here, our future work will focus on improving the reliability of W2M by reducing collisions on the wake-up signals using network coding techniques or a more robust modulation but not at the expense of higher energy consumption. In addition, we plan to enable nodes to send data in bursts to reduce the number of wake-up procedures. Finally, we want to extend our performance studies to large-scale experiments but we are experiencing difficulties to get a sufficient number of WuR prototypes.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Magno, V. Jelicic, 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, 2016.

[2] F. Aït Aoudia, M. Gautier, M. Magno, O. Berder, and L. Benini, “A Generic Framework for Modeling MAC Protocols in Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1489–1500, 2017.

[3] W. Ye, J. Heidemann, and D. Estrin, “Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks,” *ACM/IEEE Transactions on Networking*, vol. 12, no. 3, 2004.

[4] G. Anastasi, M. Conti, and M. Di Francesco, “A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks,” *IEEE Transactions in Industrial Informatics*, vol. 7, no. 1, 2011.

[5] S. Scanzio, M. Ghazi Vakili, G. Cena, C. Giovanni Demartini, B. Montrucchio, and A. Valenzano, “Wireless Sensor Networks and TSCH: A Compromise Between Reliability, Power Consumption, and Latency,” *IEEE Access*, vol. 8, 2020.

[6] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, “A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks,” in *proc. of the IEEE Wireless Communications and Networking Conference*, April 2016.

[7] A. K. Demir and S. Bilgili, “DIVA: a distributed divergecast scheduling algorithm for IEEE 802.15.4e TSCH networks,” *Springer Wireless Networks*, vol. 25, 2017.

[8] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH,” in *proc. of the 13th ACM Conference on Embedded Networked Sensor Systems*, November 2015.

[9] O. Gaddour and A. Koubâa, “RPL in a nutshell: A survey,” *Elsevier Computer Networks*, vol. 56, no. 14, 2012.

[10] 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.

[11] A. Froytlog, T. Foss, O. Bakker, G. Jevne, M. Haglund, F. Li, O. J., and G. Li, “Ultra-Low Power Wake-up Radio for 5G IoT,” *IEEE Communications Magazine*, vol. 57, no. 13, pp. 111–117, March 2019.

[12] S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, and A. V. Sheshashayee, “Wake-up Radio Ranges: A Performance Study,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6.

[13] N. E. H. Djidi, A. Courtay, M. Gautier, and O. Berder, “Adaptive relaying for wireless sensor networks leveraging wake-up receiver,” in *IEEE International Conference on Electronics Circuits and Systems*, Dec. 2018.

[14] S. L. Sampayo, J. Montavont, F. Prégaldiny, and T. Noel, “Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?” in *International Conference on Wireless and Mobile Computing, Networking and Communications*, 2018.

[15] R. Piyare, T. Istomin, and A. L. Murphy, “WaCo: A Wake-Up Radio COOJA Extension for Simulating Ultra Low Power Radios,” in *proc. of the International Conference on Embedded Wireless Systems and Networks*, February 2017.