LoBaPS: Load Balancing Parent Selection for RPL Using Wake-Up Radios

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

***Abstract*—Wake-Up Radios is an emerging technology, aiming at pushing forward the frontiers of energy efficiency without trading it off for latency nor reliability. Extending the lifetime of the nodes as much as possible is one of the main goals in Multi-hop Wireless Sensor Networks. The Routing Protocol for Low Power and Lossy Networks (RPL) is commonly used in these applications. However, there is still an open problem in its design when it comes to achieving both stability and efficient routing at the same time. In this article, we present Load Balancing Parent Selection (LoBaPS), an algorithm to select opportunistically the next hop, based on RPL. It capitalizes on the Wake-Up Radio and its always-on feature, as well as its Ultra-Low Power consumption. We compare the performance of LoBaPS with that of W-MAC, a reference protocol that uses Wake-Up Radio and supports RPL in its traditional way. The results are obtained through simulations in COOJA for a network of nodes running ContikiOS, and show that the lifetime can be improved up to 55%, while the Packet Delivery Ratio (PDR) can raise a maximum of 20%, keeping a reasonable level of latency. In addition, the network is more robust to node shutdowns and requires less control overhead.**

***Index Terms*—WSN, Wake-Up Radio, RPL, opportunistic routing, Contiki, load balancing**

I. INTRODUCTION

Every year, more devices are connecting to the Internet in different life domains such as Smart Buildings and Smart Transportation. Wireless Sensor Networks (WSN) are commonly used for such applications where there is a need for measuring some physical variable of the environment. The nodes for this goal comprise low power and resource-constrained devices with a limited distance range of communication. Traditionally, the energy consumption was controlled in these networks by some form of duty-cycle in the communication protocol at the MAC layer trading off latency for energy efficiency. In recent years the Wake-Up Radio (WuR) technology has advanced with increasing acceptance because it promises the end of this tradeoff [1]. The essentials of it are explained in Section II.

In WSN, the nodes agree on some communication protocol, so that the measured data is transported in a multi-hop fashion towards one or more collectors, called Sinks. The Routing Protocol for Low Power and Lossy Networks (RPL [2]), has been suggested by the Internet Engineering Task Force (IETF), for this matter. In RPL, the network is a Destination Oriented Directed Acyclic Graph (DODAG), where the sink is the root. One of the key features of this protocol is the rank of each node. The rank is a level of how far away a node is from the sink. In order to establish it, the nodes exchange broadcast control messages (DIO) advertising its information. To calculate the rank, RPL uses a metric, for example, ETX or the minimum amount of hops (MinHop), and an Objective Function, which translates the metric into the rank value. A commonly used Objective Function is OF0 [3] because of its stability and simple implementation. OF0 selects as the preferred parent the one with the best metric and a backup feasible successor.

This project is funded by the French National Research Agency (ANR)

However, RPL still presents some open problems: inefficient parent selection, slow recovery time after a preferred parent dies and energy bottleneck (the preferred parent consumes way more energy than the rest of its siblings limiting the lifetime of the network). ETX metric is a well-known solution to select efficient parents for reliable routes, but it presents serious issues with stability because of the recurrent parent changes. In contrast, MinHop is very stable, but might use routes with bad links [4]. In addition, the underlying duty-cycle in the MAC layer increases the latency in an effort to reduce the energy consumption, thus limiting the performance for high traffic loads.

In this article, we present LoBaPS, an approach to combine the best of both worlds: the power efficiency and always-on feature of WuR with the stability of OF0 and MinHop in RPL. The algorithm is described in Section III. Moreover, we put the focus on load balancing in order to extend the lifetime of the network. In this protocol, the nodes do not have a preferred parent. Instead, they try to wake up all the feasible successors whenever they start a communication and the actual successor is chosen by the algorithm in a decentralized way. This also provides robustness to the network that can adapt quickly to shutdowns of nodes. This is reflected in the resulting Packet Delivery Ratio. This metric, together with the latency and the lifetime of the network are compared to that of W-MAC in Section V based on the simulation framework presented in Section IV. Section VI provides a review of other relevant solutions suggested in the literature and how the present work stands out. Finally, we conclude the article and discuss future work in Section VII.

II. WAKE-UP RADIO

The Wake-Up Radio (WuR) is a secondary module connected to the main node MCU, that contains a Wake-Up

Receiver (WuRx). The distinction of this receiver is its Ultra-Low Power (ULP) consumption in listening mode between 4 and 5 orders of magnitude less than that of the traditional radios [1]. In most of the designs presented in the literature [1], the way to achieve this is using a simple On-Off Keying (OOK) modulation which requires uncomplicated circuits, and a low data rate up to 10 kbps.

As a result of this architecture, there are two communication channels. In the Main Radio (MR) channel the node uses its traditional transceiver (e.g. CC2420, etc.). In order to listen to the WuR channel, the node uses the WuRx. In contrast, to transmit on this channel, it must use some existing radio transmitter, which can be the same as that of the MR or another one, as long as it is able to modulate the signal into OOK and low data rate. The signal received on the WuRx is called the Wake-Up signal. Thanks to the ULP feature, the nodes can listen to the WuR channel continuously.

In addition, the WuR module might contain an optional sub-module to decode the data received. For this task, it is common to use a ULP microcontroller (MCU) that is placed in between the WuRx circuit and the main MCU through some sort of digital connection such as SPI or I2C [5]. In general, the information transmitted in the Wake-Up signal is the address of the destination, so that a receiver node maximizes the sleeping period of the MR and only wakes up when another node addresses it. For this reason, the WuR is especially interesting for asynchronous communications.

With this in mind, an example of how this architecture might work is given in the W-MAC protocol presented in [6]. There, the WuR is driven in the MAC layer. Whenever a node wants to communicate, it transmits a Wake-Up signal with the address of the destination, so that other nodes overhearing the WuR channel do not wake up their MR in vain. A short time (called *sync delay*) after the Wake-Up signal has been transmitted, the source transmits the data packet over the MR channel. Upon reception of this packet, the receiver sends back an ACK on the MR channel.

III. LoBaPS

The main contribution of this article is the Load Balancing Parent Selection (LoBaPS) protocol that takes advantage of the Wake-Up Radio (WuR) to select opportunistic parents in RPL. LoBaPS starts operating once RPL has converged and only supports convergecast data traffic.

A. Concept

The source of the application packet initiates the communication by transmitting a packet over the WuR channel, called Wake-Up Request (WREQ), which contains the node's own rank together with a unique application ID, as depicted in Fig. 1. All nodes in the vicinity of the sender will receive this WREQ as they are continuously listening to the WuR channel. Whenever a node receives a WREQ, it compares the received rank with its own rank, and only wakes up its main radio if the former is higher than the latter. This way, only nodes with lower rank can forward the packet, avoiding routing loops.

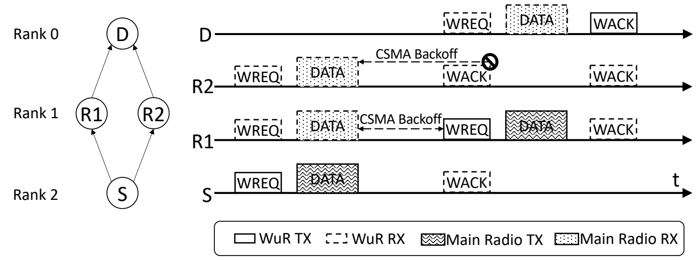


Fig. 1: Example of the algorithm in a timeline

A short time after transmitting the initial WREQ, the source sends the data packet over the main channel, turns off its main radio, and starts a timer to wait for the acknowledgment. When the sink (which is the final destination of all data packets) wakes up its main radio and receives a data packet, it sends back an acknowledgment via the WuR channel. In the case of an intermediate node, it tries to forward it by transmitting a new WREQ with its own rank. The purpose of this WREQ is threefold: to wake up next hops toward the sink and to acknowledge data reception for the sender (the third purpose is detailed in Section III-B). As a result, an acknowledgment (WACK) only differs from the WREQ that triggered its transmission by the fact that the advertised rank is lower than the one included in the WREQ.

B. WREQ collisions

A single data packet may be received by more than one parent (cf. R1 and R2 in Fig. 1). To limit collision, the CSMA layer of each forwarding node calculates a random backoff period before the transmission of the new WREQ. The node for which the backoff expires first will send a WREQ, cancelling the ongoing backoff of the other forwarders. This random backoff ensures that the feasible successors do not try to retransmit the packet at the same time generating collisions.

Collision on initial WREQ can also occur, especially when the WuR works at low data rates, because the time over the air is significant and can be longer than the one of the main data. In consequence, the channel is extremely sensitive to collisions because the transmission opportunities are very limited. Thus, a Clear Channel Assessment (CCA) function is implemented in the WuR driver and is used every time a message is transmitted over the WuR channel. When the WuR channel is sensed as busy, a collision error is passed to the CSMA layer. Although CCA is very common in traditional radio transceivers, we are part of the only few proposals investigating its usage in WuR [7]. We are convinced that such a feature is required to increase the overall network performance as supported by the results presented in Section V.

C. Cross-talk

A problem that might arise using this LoBaPS is the cross-talk, that is, when a WREQ is misread as a WACK. This can happen when multiple nodes initiate a communication at the same time, or during an ongoing communication. In order to overcome this problem, we use a unique application ID in every WREQ. This ID is set by the original source and kept unmodified by intermediate nodes until reaching the sink. Furthermore, whenever a node initiates a new communication, it uses a different application ID than the previous one. Therefore, a relay can distinguish between a new WREQ or the WACK of one of its previous transmission. For our implementation, we assign a batch of unique application IDs that can be used for each node to start a communication.

D. Retransmissions avoidance

Load balancing can only be achieved if the forwarders are well distributed across the network and if the duplicated packets are reduced as much as possible. If a source would miss an acknowledgment, then the CSMA layer would try to transmit the packet again waking up all its feasible successors one more time even if one of them have already forwarded it. In order to reduce this problem, each node keeps a list of recently viewed application IDs and a list of forwarded application IDs. The first list is intended to avoid waking up the main radio and waste power listening to a packet that has already been handled by another node. On the other hand, the second list keeps track of the application IDs that have actually been transmitted by this node. Whenever a node receives a WREQ with an application ID that is currently in this list, it will immediately transmit a WACK with its own rank to let the source know that this packet has already been handled and that it must refrain from retransmitting it. We call this specific WACK a WuR Duplicated ACK (WDA). Both lists are cleaned periodically to avoid outdated values.

In addition, when the winner of the forwarding competition (the node with the shortest backoff period) transmits the WREQ, the rest of the competitors overhear this signal and refrain from retransmitting, thus avoiding duplicated packets.

IV. Simulation framework

In the literature, GreenCastalia or COOJA are commonly used to simulate a network of nodes supporting the Wake-Up Radio. We use an extension of the latter one, called WaCo [6] because it reproduces the actual firmware that runs on real devices. On top of that framework, we added some new features for this work.

A. Simulation setup

A summary of the simulation parameters is given in Table I. We use the default values in ContikiOS for other parameters.

The energy consumption is calculated with the help of Powertrace. This tool was extended in this contribution to support the Wake-Up Radio. The electric values required are taken from the Sky Mote datasheet [8] and WaCo [6].

Additionally, the simulations are performed in a triangular grid topology as depicted in Fig. 2. Here, the nodes are at a maximum of 2 hops away from the sink and with a node density (i.e. number of nodes per unit area) such that each leaf can have between 2 and 7 feasible parents. In order to simulate its battery lifetime, each node keeps track of the energy consumed and when it reaches some maximum level (defined as a parameter) the node is killed. When this happens there are two different scenarios that follow. First, some children might be temporarily unreachable, but after the RPL repair mechanism is triggered (which is better described in the following Section IV-B), a new parent can be found and the network graph continues connected. Second, some nodes might be left far away from any other one, becoming absolutely unreachable, and no mechanism can get the network graph connected again. The simulation is stopped immediately and only when the second scenario is found. This way, it is possible to analyze the behavior of the network after the first node dies and throughout the process of parent changes.

The comparison of the WuR architecture against duty-cycled MAC protocols has already been done in previous

TABLE I: Simulation parameters

Parameter	Value
Number of nodes	15
Repetitions of each simulation	100
MAC layer	CSMA (Contiki version)
CSMA minBE	3
CSMA maxBE	5
CSMA maxBackoff	4
CSMA maxRetries	3
Network layer	uIPv6
RPL Mode	No downward routes (MOP 0)
RPL Objective Function	OF0 [3]
RPL Metric	Minhop
RPL MAX_FAILED_PACKETS	4
Packet generation period	10 s
WuR packet length	16 bits
WuR data rate	10 kbps
Main data packet length	80 bytes
Main data ACK packet length	5 bytes
Main radio data rate	250 kbps
Main Node	Sky mote [8]
WuR HW prototype	[5]
WuR Supply Voltage	1.8 V
WuR TX current	16 mA
WuR RX current	80 uA
WuR idle listening power consumption	1.944 μW
Main Radio medium model	UDGM
WuR Radio medium model	UDGMConstantLoss
Main radio RX success ratio	80%
WuR RX success ratio	80%

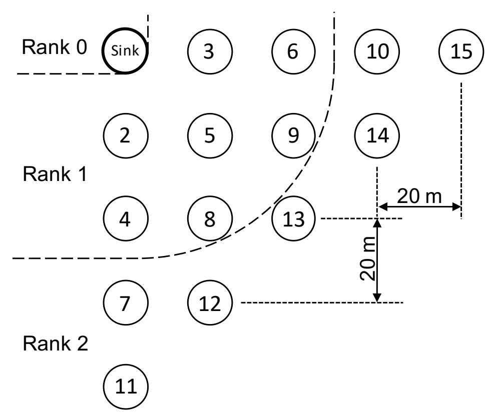


Fig. 2: Test topology

works [1] [9] [6] [10] and shows the superiority of the first one if some conditions are guaranteed (e.g. medium or low traffic loads and short distance range applications).

Both LoBaPS and W-MAC are implemented in two versions: regular and without acknowledgments (identified by the suffix 'NA').

B. Optimized W-MAC

This protocol is based on W-MAC [6], a straight forward utilization of the WuR, already described in section II. In this work, the parameters of this protocol have been optimized (sync delay, reception window timeout, etc.) and a CCA capability has been introduced to avoid collisions. In addition, the length of the Wake-Up signal has been fixed to 2 bytes, which complies with the short address of 16 bits in IEEE 802.15.4. We use the default values in that standard for the CSMA algorithm on top of this layer.

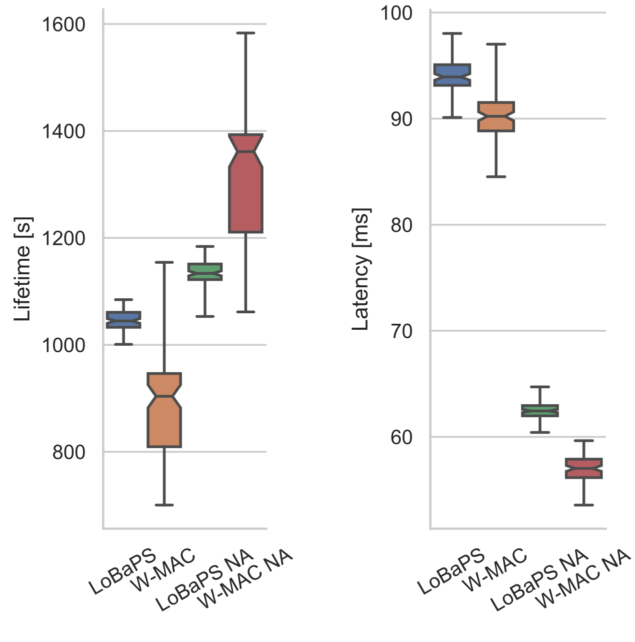
This protocol uses RPL with Objective Function 0 and MinHop metric. In RPL OF0 RFC [3] there is no clear explanation on how to detect that the preferred parent of a node is no longer available. In the general implementations, when a preferred parent dies, the node does not realize it, and there is no mechanism to trigger a parent change. However, the RPL OF0 RFC provides a *backup feasible successor* that can be used whenever this happens. For this reason, we implement the parent change trigger when a fixed number (MAX_FAILED_PACKETS) of communication attempts fail consecutively. Notice that a communication attempt includes all the retries at the CSMA layer. This means that if the maximum amount of retries at the CSMA layer is 3 for example, and the MAX_FAILED_PACKETS is 4, then the parent change will be triggered after $3 \cdot 4 = 12$ acknowledgments not received consecutively. As soon as this happens the preferred parent is removed from the parent set and the backup feasible successor is used instead. Then, if at some point the backup parent also fails MAX_FAILED_PACKETS times, the parent set is cleaned and a RPL local repair is issued.

Another version of W-MAC is implemented without the use of acknowledgments. In this case, the receiver does not send an acknowledgment back, and the source does not trigger any mechanism for retransmissions in the CSMA layer.

V. Results

For the purpose of studying the benefits of the proposed algorithm, we analyze the network lifetime (that is the time elapsed when the first node dies), the average Packet Delivery Ratio (PDR), the average end-to-end latency, the remaining battery in each node when the first one dies, and the control overhead of the protocol. For the PDR, we analyze the evolution of its value over time through the whole simulation (that is, while the network is a connected graph).

We generate a notch plot for the lifetime and latency results. This shows the median value, boxes with an IRQ (interquartile range) of 25th to the 75th percentile, whiskers for the maximum and minimum values, and the confidence intervals of 95% ensuring that our measurements are statistically significant. Regarding the remaining battery results and the PDR over time, we show a bar plot and a line plot respectively, both with the mean value and the 95% confidence intervals.



(a) Network lifetime (b) End-to-end Latency

Fig. 3: Lifetime and Latency metrics

A. Lifetime

The network lifetime results are shown in Fig. 3a. The large interquartile range in the W-MAC protocols shows high variability, while the small ones in LoBaPS ones denote its precision. This is because the result depends on the selection of preferred parents that is taken at random at the beginning of the simulation depending on which nodes sent the first DIO. When most of the leaves select the same parent, then this node is overwhelmed and consumes most of the network energy, reducing its lifetime. On the other hand, when the preferred parent of each leaf is a different one, then the network is balanced.

In the best case, LoBaPS achieves a 55% better lifetime than W-MAC, and the improvement in mean values is of 17%. The reason why is that the network balancing does not depend on the initial DIOs exchange. All feasible successors of a node can compete to be its parent every time the node transmits.

At the same time, in the versions of the protocol without ACKs we can see that there is an inversion: W-MAC NA achieves a longer lifetime in median value, though with less precision. This is because in LoBaPS every feasible successor wakes up and listen through the Main Radio wasting energy.

B. Remaining battery

The performance of the load balancing algorithm can be best described by the remaining battery of the rank neighbors at the precise moment when the first node dies (this is the time

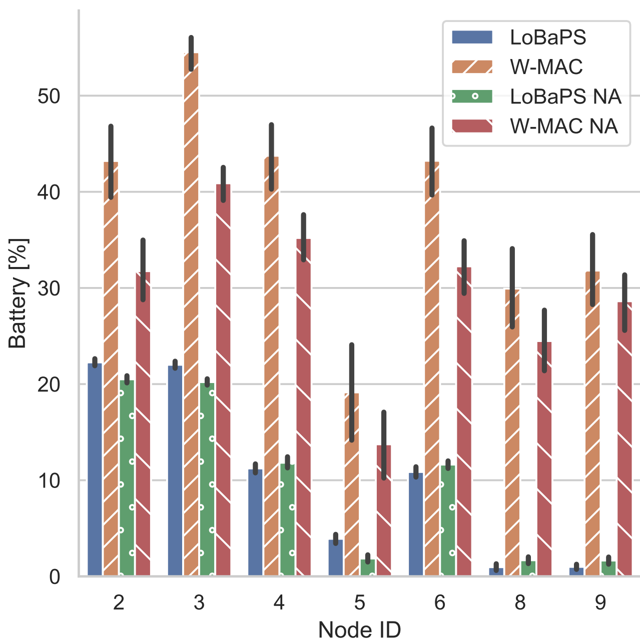


Fig. 4: Remaining Battery of nodes in the same rank when the first node dies

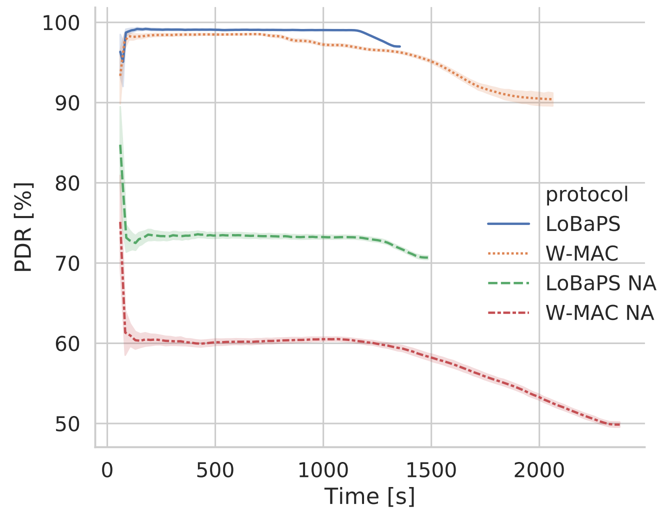


Fig. 5: Packet Delivery Ratio

of the network lifetime). As we have seen in Section V-A, in W-MAC that precise moment (around 850 s) is sooner than in LoBaPS (around 1050 s), which is equivalent to say that the lifetime is shorter. This explains why in Fig. 4 we can see that the neighbors in the same rank as that first node have still a high amount of remaining energy that was not exploited by the network. On the contrary, in LoBaPS, all rank neighbors arrive with a small amount of remaining energy to the end of the network lifetime, proving the load balancing. The network is able to consume the energy budget more efficiently,

C. *Packet Delivery Ratio*

Fig. 5 shows the evolution of the PDR over time. There it is possible to see the decline of the PDR when the first node dies. In the case of W-MAC, this is between 750 s and 1200 s, whereas in LoBaPS it is around 1100 s. We can see that in W-MAC protocols the PDR goes down fast and with high variability. On the contrary, LoBaPS versions provide good stability during the network lifetime and a precise and controlled decline slope. The final PDR can be between 2% and 20% better in LoBaPS because of its robustness to parents dying. In contrast, in W-MAC when a preferred parent dies, it only selects its backup feasible successor after some fixed amount of packets have failed to be delivered by a child. Then, when that backup parent dies, the RPL Local Repair mechanism is triggered only after that fixed amount of packets fail again.

The mean steady-state PDR of LoBaPS is slightly better (less than 1%) than that of W-MAC. However, as it happens with the lifetime, LoBaPS is more precise, while the W-MAC results are more variable.

D. *End-to-end Latency*

In W-MAC and LoBaPS, the latency is mainly dominated by the CSMA backoff period. This value can only be incremented before transmitting when the MAC or physical layers detect that there will be a collision if the packet was transmitted and then passes a *collision error* to the CSMA layer, which increments the backoff exponential accordingly. The latency is also impacted by the amount of retransmissions required to deliver a packet correctly. In LoBaPS there are more Wake-Up signals occupying the channel because of the WACKs and WDAs not present in W-MAC.

Fig. 3b shows that the latency is slightly worse in LoBaPS protocols (3% higher) because of the higher chances of collisions, provided by a larger amount of Wake-Up signals. In addition, the last ACK in a communication is sent by the sink using the WuR instead of the Main Radio, thus with a lower data rate. As a consequence, the time over the air of the packet is longer, incrementing the end-to-end latency.

On the other hand, when the protocols do not expect acknowledgements, they also do not generate retransmissions. As a matter of fact, the packet can only be delivered correctly in one shot, that is without retransmissions. In consequence, the latency is smaller for this type of protocols.

E. *Control overhead*

The control overhead coefficient for each protocol is calculated with the following formula:

$$c = 100\frac{\#\, network\ control\ packets}{\#\, app\ packets\ at\ the\ sink}$$

The results for LoBaPS and W-MAC are 8% and 13% respectively. Our algorithm relies on the RPL graph built at the initialization phase of the network and does not need to pull any control mechanism if some parent fails because in general there is another one able to forward the packet, and all the parents die approximately at the same time.

VI. Related work

W-MAC [6] was explained in Section II. This solution is the only one that has been proved to work along with RPL in the literature. While it achieves a great power efficiency, latency, and reliability compared to the duty-cycled approach, it does not fix the main problems of RPL, summarized in Section I.

In [10], the authors presented OPWUM to allow opportunistic forwarding based on a given metric. This protocol uses backoff timers, according to that metric, to delay the CTS of all the neighbors that receive an RTS. Then, the source transmits an ATS so that every neighbor knows that another one won the competition, and only then it transmits the data packet on the Main Radio channel. In addition, it does not use CCA and it has not been tested with RPL.

GreenRoutes and WHRAP are presented in [11] and [12]. Both protocols are cross-layer routing solutions on its own, where the network is initialized by the sink with broadcast control packets so that all the nodes in the network know its hop-distance to it. Then, the energy is taken into account to opportunistically select the next relay, obtaining a great performance focused on energy harvesting systems. Furthermore, the communication for each link is established by a sequence of RTS, CTS, and ATS packets, and it does not consider the use of CCA for the WuR.

Ghose et al. presented BoWuR in [13], CCA-WuR, CSMA-WuR and ADP-WuR in [7]. In those works, they demonstrate the importance of using CCA and backoffs to improve the WuR performance under high traffic loads. Nevertheless, it is only targeted to single-hop networks and there is no guideline on the way to extend it to multi-hop scenarios.

To the best of our knowledge, LoBaPS is the first work that combines the best features of WuR and RPL to overcome some of the routing challenges (presented in Section I) in multi-hop scenarios. Together with [7], it is one of the first ones to point out the need for a CCA feature in the WuR prototypes. Moreover, it is the first one to provide it in simulations.

VII. Conclusions and future work

This article introduces LoBaPS a load balancing parent selection algorithm. The main idea is to allow all feasible successors to compete for a packet forwarding when a node transmits a packet, taking advantage of the always-on feature of the Wake-Up Radio. At the same time, it mitigates the duplicated packets and main radio listening energy consumption. We showed that it overcomes the single point of failure problem at the preferred parent of traditional RPL with Objective Function Zero and MinHop metric. This way, it also extends the network lifetime up to 55% by consuming the battery of the feasible successors in a balanced way. Furthermore, we found that the nature of this mechanism is more precise, providing more reproducibility than the traditional implementation of RPL. This together with an improvement of the Packet Delivery Ratio of up to 20% gives more reliability to the network infrastructure.

The main drawback of the proposed algorithm is the amount of energy wasted in listening mode when all the feasible successors wake up its main radio, limiting the network lifetime. Although in the long run, the parents with the best quality links probably win the competition more often than parents with bad links, nothing ensures that the most reliable route is chosen. This is another limitation of this work and we plan to investigate a solution in the future.

In our next steps, we plan to extend this algorithm to balance arbitrary metrics and not only packet transmissions as well as experiments on real devices.

Acknowledgments

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

References

[1] 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.

[2] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," Internet Requests for Comments, RFC Editor, RFC 6550, March 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6550.txt

[3] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," Internet Requests for Comments, RFC Editor, RFC 6552, March 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6552.txt

[4] O. Iova, F. Theoleyre, and T. Noel, "Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2013, pp. 2098–2102.

[5] 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, April 2016.

[6] R. Piyare, T. Istomin, and A. L. Murphy, "Waco: A wake-up radio cooja extension for simulating ultra low power radios," in *EWSN*, 2017.

[7] D. Ghose, F. Y. Li, and V. Pla, "MAC Protocols for Wake-Up Radio: Principles, Modeling and Performance Analysis," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2294–2306, May 2018.

[8] "T-mote sky datasheet." [Online]. Available: http://www.eecs.harvard.edu/ konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[9] S. L. Sampayo, J. Montavont, F. Prégaldiny, and T. Noël, "Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?" in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, pp. 1–8.

[10] 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

[11] S. Basagni, V. D. Valerio, G. Koutsandria, and C. Petrioli, "Wake-Up Radio-Enabled Routing for Green Wireless Sensor Networks," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–6.

[12] S. Basagni, V. D. Valerio, G. Koutsandria, C. Petrioli, and D. Spenza, "WHARP: A Wake-Up Radio and Harvesting-Based Forwarding Strategy for Green Wireless Networks," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct. 2017, pp. 257–265.

[13] D. Ghose and F. Y. Li, "Enabling Backoff for SCM Wake-Up Radio: Protocol and Modeling," *IEEE Communications Letters*, vol. 21, no. 5, pp. 1031–1034, May 2017.