HAL
open science

eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT

Sebastian Lucas Sampayo, Julien Montavont, Thomas Noël

To cite this version:

Sebastian Lucas Sampayo, Julien Montavont, Thomas Noël. eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT. 18th International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now), Oct 2019, Luxembourg, Luxembourg. 10.1007/978-3-030-31831-4_27 . hal-03698412

HAL Id: hal-03698412

https://hal.science/hal-03698412v1

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.

eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT

Sebastian L. Sampayo, Julien Montavont, and Thomas Noël

ICube Laboratory, University of Strasbourg, 67412 Illkirch, France
{sampayo, montavont, noel}@unistra.fr

Abstract. In an effort to extend the lifetime and reliability of multi-hop wireless sensor networks we recently presented LoBaPS, a protocol to select opportunistic parents and achieve load balancing. This algorithm takes advantage of the wake-up radio for its ultra-low power consumption and always-on feature. Moreover, it overcomes an open problem in the routing layer: achieving both stability and efficient parent selection at the same time. However, the random load balancing strategy and the energy wastage in listening mode still limits the network lifetime. In this article, we present eLoBaPS, a significant modification of LoBaPS that distributes better the energy among the parents improving around 17% the lifetime of the network towards the ideal case. In a nutshell, the next hop is selected in a decentralized way and it is the parent that issues a shorter back-off period before attempting to retransmit. In addition, the nodes overhear all the traffic in the wake-up radio channel and adapt the protocol parameters to the current state of the battery of the neighbors. We perform simulations with a network of ContikiOS nodes running eLoBaPS, LoBaPS and W-MAC, a reference protocol that uses the wake-up radio.

Keywords: WSN · Wake-Up Radio · opportunistic routing · Contiki · load balancing

1 Introduction

The global smart city market was valued at five hundred billion dollars in 2017 and is projected to reach two thousand billion by 2025 [1]. Wireless Sensor Networks (WSN) with low power and resource-constrained devices are generally used for those IoT applications. The energy consumption was traditionally controlled in these networks by some form of duty-cycle in the communication protocol at the MAC layer trading off latency for energy efficient operations. Recently, the Wake-Up Radio (WuR) technology has advanced with increasing acceptance as a promise to the end of this tradeoff [7]. Its fundamentals are explained in Section 2.

On top of that, the nodes need to compute a path towards a collecting station (called the sink) in order to deliver their data in a multi-hop fashion. This is the role of the routing protocol generally operating at the layer 3 of the network

stack. When the system starts operating, the nodes exchange control packets to generate a routing structure in order to reach the sink, and eventually, other peer-nodes. In general, the implementation of this structure is based on a tree, where each node defines a preferred parent to forward its data packets towards the sink. However, it is difficult to achieve both stability and efficient parent selection at the same time in these routing protocols. A simple implementation can provide stability but at the cost of using sometimes bad links, thus degrading the performance. On the other hand, a more complex protocol can improve the parent selection for good links, but usually, this leads to an increase in the number of control packets and degraded stability, since each node may change its preferred parent very frequently. 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.

Load Balancing Parent Selection (LoBaPS) [9] has been presented in previous works leveraging the power efficiency and always-on feature of WuR. It proposes a solution to the tradeoff between stability and efficient parent selection. Once the routing structure is built, the preferred parent is not exclusively used to forward data packets. Instead, all available feasible parents compete in order to share the load. This approach presents a first step into load balancing with WuRs in order to extend the network lifetime. However, LoBaPS balances the load in a random way and therefore can not achieve the best energy balancing among nodes. LoBaPS is described in Section 4.1.

In this article, we present Energy LoBaPS (eLoBaPS), an improvement over LoBaPS that takes a step forward into the ideal energy balancing in WSN. In this protocol, the backoff period is proportional to the battery consumed by each parent. In addition, the nodes that are consuming a lot stop competing for a while and let other feasible forwarders spend their batteries. In Section 3 we provide a review of the related publications and how the present work stands out. Our contribution, eLoBaPS, is described in Section 4.2. The energy savings achieved are reflected in the resulting lifetime that is compared to that of W-MAC and LoBaPS. In addition, we look at the packet delivery ratio of the network over time to compare the stability and final decline of the operation performance, where eLoBaPS has a longer stable operation and a shorter decline. This metric, together with the latency and the productivity of the network are compared to that of W-MAC and LoBaPS in Section 6 based on the simulation framework presented in Section 5. Finally, the conclusions of the article are presented in Section 7.

2 Wake-Up Radio

The Wake-Up Radio (WuR) is a secondary hardware module that contains a Wake-Up Receiver (WuRx) and is connected to the microcontroller (MCU) of the main node. This receiver is highlighted by the Ultra-Low Power (ULP) consumption in listening mode around 5 orders of magnitude less than that of the traditional radios [7]. This is usually achieved by using a simple On-Off Key-

ing (OOK) modulation and a low data rate up to 10 kbps [7]. As a result of this architecture, there are two communication channels. On the one hand, the communication in the Main Radio (MR) channel is performed by the traditional transceiver of the node (e.g. CC2420). On the other hand, the node uses the WuRx to continuously listen to the WuR channel thanks to the ULP feature. However, the transmissions on the WuR channel do not involve a dedicated chip, but generally use the same as that of the MR provided that it supports OOK modulation and low data rate.

Having said that, the W-MAC protocol presented in [8] is an example of how this architecture might work. In that case, the WuR is controlled at the MAC layer. To initiate the communication, a node sends a message (called thereafter wake-up signal) over the WuR channel to wake up the MR of the layer 2 destination. Shortly after, the source sends the data packet over the MR channel. Upon reception of this packet, the receiver sends back an ACK on the MR channel. As a result, there is no need of duty-cycling the activity/sleeping period nor continuous synchronization as destinations will only wake up their MR on-demand via the WuR channel.

Optionally, the WuR hardware block might contain a module to decode the wake-up signal received. This task is performed normally by a ULP microcontroller (MCU) that is placed between the WuRx circuit and the main MCU through some sort of digital connection such as SPI or I2C [6]. It is common to transmit the address of the destination node in the wake-up signal so that a receiver node maximizes the sleeping period of the MR and only wakes up if there is a match between its own address and the received wake-up signal. This is the reason why the WuR is particularly interesting for asynchronous communications.

3 Related work

eLoBaPS is an improvement of LoBaPS [9], providing a solution to its main drawbacks: random load balancing and energy wastage in listening mode. To the best of our knowledge, LoBaPS was the first work leveraging the WuR to overcome some of the routing challenges (presented in Section 1) in multi-hop scenarios. It is a further step towards the energy load balancing in WSN with WuR based on the work of W-MAC [8], OPWUM [3], WHARP [4] and Green-Routes [5], briefly described below.

In Section 2 we briefly described W-MAC [8]. Although it achieves a great power efficiency, latency and reliability in comparison to the duty-cycled approach, it does not tackle the problems of the routing layer summarized in Section 1. OPWUM [3] allows opportunistic forwarding using the WuR and is based on the well-known RTS/CTS paradigm. A source can reserve the MR channel by exchanging 3 broadcast messages over the WuR channel. However, OPWUM was evaluated at a very small-scale (with 4 nodes - 1 source, 1 destination, and 2 relays). We already showed [9] that increasing the WuR channel usage is prone to collisions and errors because transmission opportunities are very limited due

to the time over the air required to transmit wake-up signals (most of the current prototypes work at 1kbps). By contrast, eLoBaPS minimizes the usage of the WuR channel (if the path between a source and the sink includes n nodes, eLoBaPS sends n wake-up signals instead of $3(n-1)$ for OPWUM).

In [4] and [5], the authors presented WHARP and GreenRoutes. In those solutions, the next relay is opportunistically selected by taking into account the energy, obtaining great performance. However, those schemes are cross-layer solutions including their own routing solution. In consequence, they are less generic as it is not possible to use a well-known routing protocol. In addition, they use the RTS/CTS paradigm on the WuR channel increasing the channel utilization.

4 LoBaPS and eLoBaPS

4.1 LoBaPS

LoBaPS was presented in [9] featuring the Wake-Up Radio (WuR) to select opportunistic parents in the layer 3 of the communication stack. This protocol operates only under convergecast data traffic and when the routing layer has converged so there is a stable topology where each node knows its own metric with respect to the sink. For example, this metric could be how many hops away it is from the sink. We are going to call this metric r.

Whenever a source node generates an application packet, it initiates the communication by transmitting a packet over the WuR channel: the Wake-Up Request (WREQ), as depicted in Fig. 1. This packet contains the metric r of the transmitting node. The nodes are continuously listening to the WuR channel, so all the nodes inside the communication range of the sender will receive this WREQ. Upon reception, the received metric is compared with the r of the node, so that only nodes that have a better metric (i.e. nodes that are closer to the sink) wake up their MR, avoiding routing loops. Shortly after the transmission of the initial WREQ, the data packet is sent by the source on the MR channel. Then the source turns off its MR and starts a timer waiting for the acknowledgment. When the sink wakes up its MR and receives a data packet, it sends back an acknowledgment via the WuR channel. Conversely, when an intermediate node receives the data packet, it tries to forward it by transmitting a new WREQ with its own metric r. This WREQ has three purposes: to wake up next hops towards the sink, to acknowledge data reception for the sender, and to stop other competitors.

A data packet may be received by several nodes, since a single node may have multiple parents (cf. R1 and R2 in Fig. 1). In order to mitigate collisions, a random backoff period is calculated by the Carrier Sense Multiple Access (CSMA) layer of each relay before the transmission of the new WREQ. The first one whose backoff expires will send a WREQ and cancel the ongoing backoff of the others.

Given that the WuR works at low data rates, the time over the air for wake-up signals is significant and can be longer than the one of the main data. This

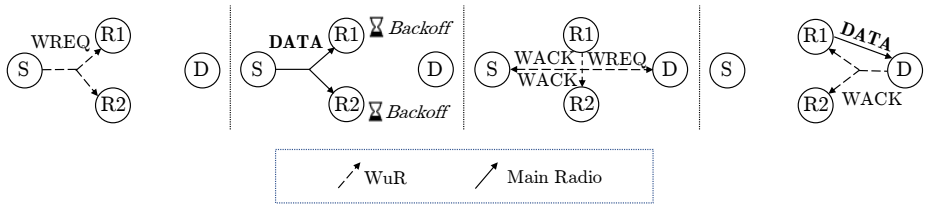

Fig. 1. Example of LoBaPS/eLoBaPS operations

way, transmission opportunities over the WuR channel are very limited, which results in a high sensitivity to collisions. For this reason, LoBaPS includes a Clear Channel Assessment (CCA) function in the WuR driver. When a node wants to transmit, it calls first the CCA function, and if the WuR channel is not clear a collision error is passed to the CSMA layer.

4.2 eLoBaPS: Improved load balancing

The main problems with LoBaPS are the way in which the load is balanced and the energy wastage in listening mode. The load is balanced randomly, so it is not the best effort towards energy efficiency. On top of that, there is a significant amount of energy wasted in listening mode when all the feasible successors wake up their MR, limiting the network lifetime. In this article, we present eLoBaPS, an improvement of LoBaPS that increases the network lifetime and provides better energy balancing. In this extension, the general behavior is also described by Fig. 1. In this case, the backoff period is proportional to the energy consumed by the node, so that nodes with more remaining battery have more chances to win the competition. The first approach in order to calculate a backoff period aware of the energy consumption is to make it directly proportional to the amount of consumed battery percentage:

$$B_j(t) = Ke_j(t) + C \tag{1}$$

where $e_j(t)$ is the energy percentage consumed by the node j at time t, and K is a constant parameter to adjust the units to milliseconds. A small random contention window C uniformly distributed with range $[0, T_c]$ is added to mitigate the case where more than one node has the same amount of energy. However, the battery discharges as the time goes by, so $e_j(t)$ is proportional to t. In consequence, the backoff period will increase over time as the battery discharges. Notice that the backoff period adds latency to the protocol, so it would increase the end-to-end delay of the application. In order to keep it stable as the battery discharges, we came up with a slight modification of Eq. 1:

$$B_j(t) = K[e_j(t) - e_{dj}(t)] + C \tag{2}$$

where $e_{dj}(t)$ is the desired energy consumed at time t. This variable is such that if all the relays consume this energy, the load is balanced and the network

lifetime is maximized. In order to estimate it, the nodes include their energy consumption in all the packets sent on the WuR channel (WREQs and WACKs) so that they overhear the current energy of all their neighbors. Then the value is estimated for node j with a metric r as:

$$e_{dj}(t) = \min_{i \in R}\{e_i(t)\} \tag{3}$$

where R is the set of all nodes i with a metric of r.

Although this may balance the energy by controlling the transmissions, it is not reducing the energy consumed by the listening mode every time a node requests several relays to wake up and listen to the packet (see node R2 in Fig. 1). With this in mind, another feature is included in the algorithm to reduce the listening mode energy: if the current energy consumed by the node is above a certain threshold on top of the desired energy consumption, then the node does not wake up and listen to the MR channel whenever it receives a WREQ. However, this feature can create problems when a node is the only parent possible for some nodes for example. In such a situation, the packet will be delayed until the parent saves enough energy to keep up with $e_d(t)$. For these reasons, this threshold should be chosen carefully to not degrade the performance. In this work, it has been set to the energy consumed by the node that has consumed the most among all its neighbors in R, that is:

$$e_{thresholdj}(t) = \max_{i \in R}\{e_i(t)\} \tag{4}$$

Fig. 2, Fig. 3, Fig. 4, and Fig. 5 may help to understand the overall behavior of the protocol. Fig. 2 shows the overall finite state machine of a generic node running the eLoBaPS protocol. Besides, Fig. 3, Fig. 4, and Fig. 5 describe the internals of each mode. When the application layer of the node wants to send a message, it issues a transmission request to the lower layers. The CSMA layer receives this request and performs the algorithm described in Fig. 3, calling the routine of the TRANSMIT mode, detailed in Fig. 4. Notice that the COLLISION output of the TRANSMIT routine is a flag to prevent from a collision. In such a case, the TRANSMIT routine does not get to transmit the DATA on the MR. This is the classic behavior of the CSMA protocol. Its implementation in ContikiOS was not significantly modified. The values of the constant parameters used in our implementation can be found in Table 1. On the other hand, when a node receives a WREQ, an interrupt is triggered and the RECEIVE mode is activated, following the steps in Fig. 5.

5 Simulation framework

For the performance evaluation of eLoBaPS, we used WaCo [8]. This is a simulation framework that is an extension of ContikiOS and Cooja that reproduces the firmware that runs on real devices. WaCo is publicly available and has been validated with a WuR prototype.

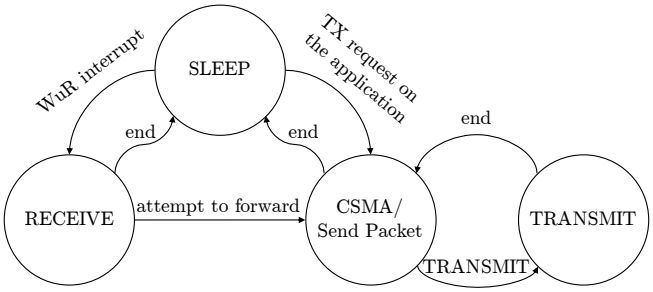

Fig. 2. Finite state machine of the entire system

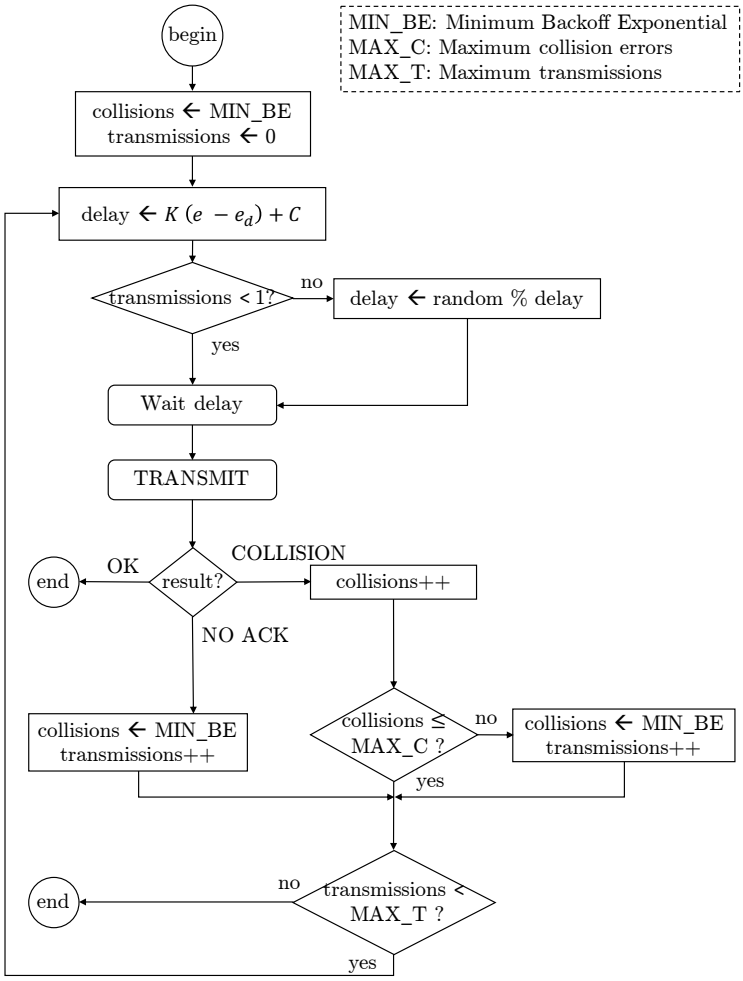

Fig. 3. CSMA mode flowchart

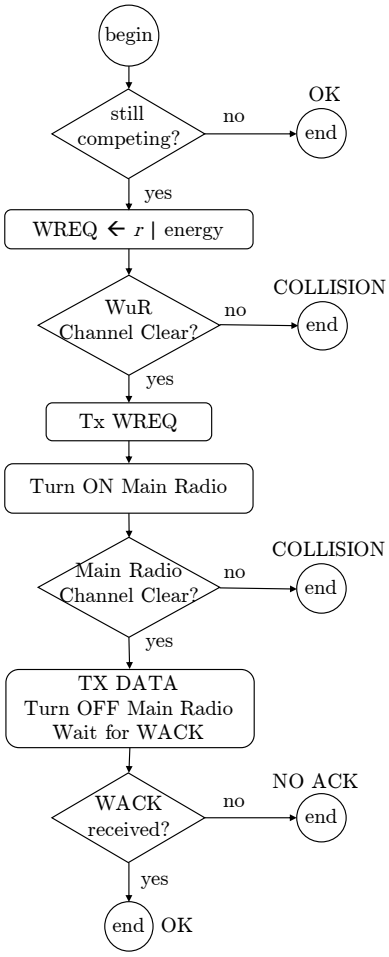

Fig. 4. TRANSMIT mode flowchart

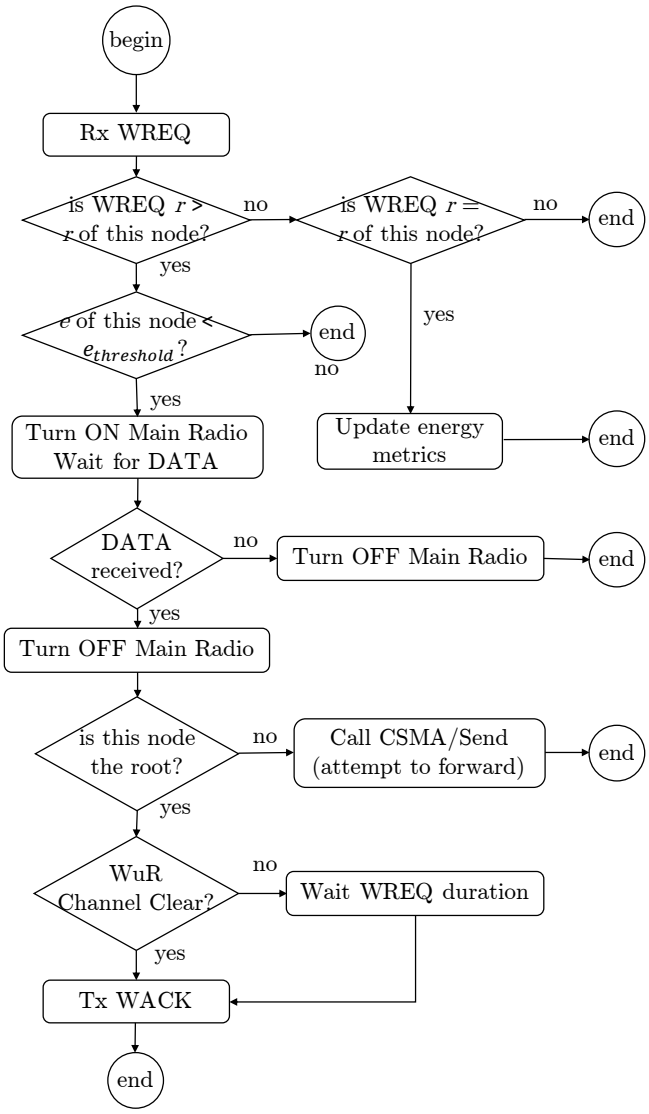

Fig. 5. RECEIVE mode flowchart

Table 1. Simulation parameters

Parameter	Value
Number of nodes	15
Repetitions of each simulation	50
eLoBaPS T_c	30 ms
eLoBaPS K	11.6 ms
MIN_BE	3
MAX_C	7
MAX_T	4
Inter Packet Interval (IPI)	1, 5, 10, 60 s
WuR packet length	16 bits
WuR data rate	10 kbps
Main radio RX success ratio	80%
WuR RX success ratio	80%

5.1 Simulation setup

Table 1 depicts the main simulation parameters. The simulations are carried out in a triangular grid topology as depicted in Fig. 6. The nodes are not more than 2 hops away from the sink and each leaf can have between 2 and 7 feasible parents. In this figure, we can see the links, represented by arrows, of each node to all its feasible next hops. However, for legibility purposes, dot line arrows are the links to the sink of the nodes that are just 1 hop away from it, and solid line arrows are the links to the parents of the nodes that are 2 hops away from the sink. Notice that some nodes have more chances to listen and forward packets since they have more children, in particular nodes 5, 8, and 9. To simulate the battery lifetime, each node keeps track of the energy consumed with a linear model, so that it is killed when it reaches some threshold level. This value is defined so that the sink receives around 1000 packets when the first node dies. At this point, we find two different scenarios that follow. First, some children might be temporarily unreachable (in the case of W-MAC with a routing protocol that is based on a preferred parent), but after a repair mechanism is triggered, a new parent can be found and all the living nodes can reach the sink again. 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 finishes immediately and exclusively when the second scenario is found. This way, we can analyze the behavior of the network after the first node dies and during the process of parent changes.

We use an optimized version of W-MAC that supports the Routing Protocol for Low Power and Lossy networks (RPL) with Objective Function 0 and MinHop metric, as well as CCA capability in the WuR channel. In RPL OF0 RFC [11] there is no clear explanation on how to detect that the preferred parent of a node is no longer available. For this reason, we implement the parent change trigger when a fixed number of communication attempts fail consecutively (which in our implementation is set to 4). This means that every time a

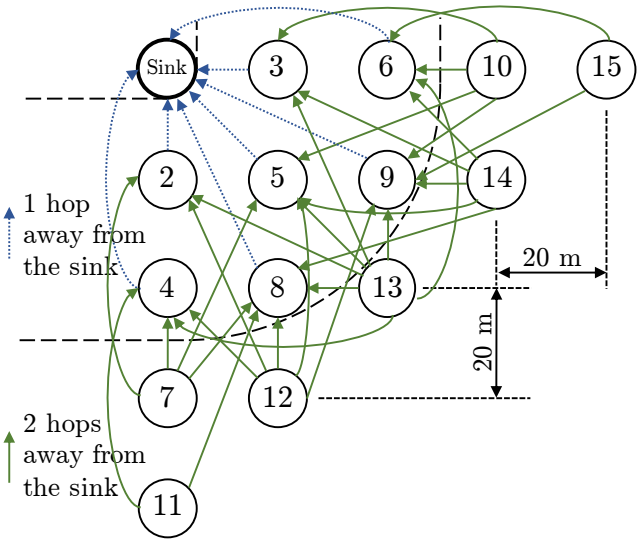

Fig. 6. Test topology

parent dies, it takes some time to the W-MAC solution to return to its correct operations. This routine is triggered after MAX_T transmission attempts on the CSMA layer.

6 Results

6.1 Lifetime

Fig. 7 shows the results for the relative network lifetime under different traffic loads with respect to the lifetime of W-MAC. The network lifetime is defined in this work as the time elapsed when the first node dies. We show the relative

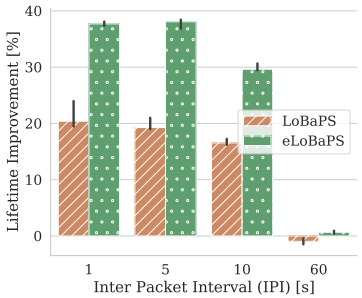

Fig. 7. Lifetime improvement over W-MAC for different traffic loads

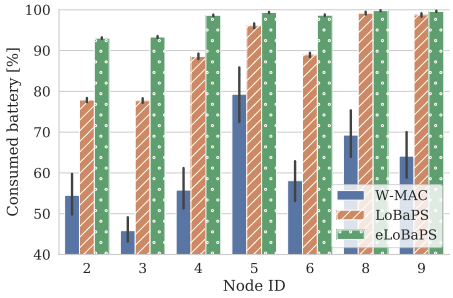

Fig. 8. Consumed battery of nodes at 1 hop from the sink when the first node dies

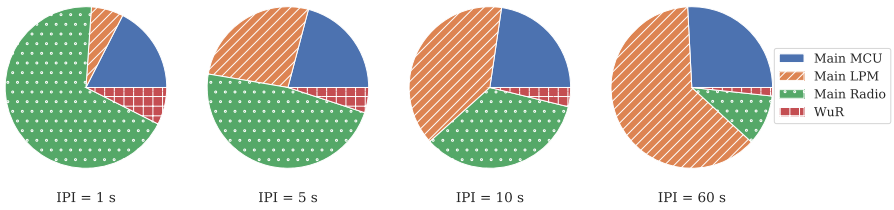

Fig. 9. Energy profile for different IPIs for the W-MAC protocol

value of the lifetime so that the results can be appreciated on the same scale for all protocols and all values of the Inter Packet Interval (IPI). However, to get a rough idea of the absolute values when using two AA batteries, the lifetime of W-MAC is 90 days when the IPI is 1 s and 3 years when the IPI is 60 s in mean values.

The Inter Packet Interval (IPI) between the generation of application packets has been varied from 1 s (high traffic) up to 60 s (low traffic) which are typical values in smart cities applications [10]. The bar plot shows the median values for each scenario over the 50 repetitions and the confidence intervals of 95%. The results show the superiority of eLoBaPS over its predecessor LoBaPS (up to 17%) and the reference W-MAC (up to 40%) in all scenarios. Studying the maximum and minimum outcomes for all the repetitions (not shown in the figure), we can affirm that the lifetime improvement can actually go up to 77% in the best case of eLoBaPS compared to the worst case of W-MAC. This is because of the two main features of the protocol: the energy consumption due to packet transmissions is well distributed among all the feasible parents and the nodes that are excessively consuming energy turn off their MR till they keep up with the energy consumption of their neighbors.

At the same time, it is interesting to remark that in low traffic scenarios almost all the protocols present very small differences in the resulting lifetime. The reason behind this is that as the IPI increases, the impact of the radio communications on the overall energy consumption decreases. Fig. 9 supports this fact by depicting the contribution to the overall energy consumption of the different power states of an average node running W-MAC: main MCU in active mode, main MCU in Low Power Mode (LPM), main radio, and wake-up radio. The radio contributions (Main Radio and WuR) comprise transmission, reception and listening for each one. In this figure, it can be noticed that the contribution of the radio modes (Main Radio and WuR) is the most significant to the total only for high traffic scenarios (1 s IPI). On the contrary, in low traffic scenarios, the major contribution to the overall energy is just the LPM, that is, because of the silent power consumption when the node is sleeping.

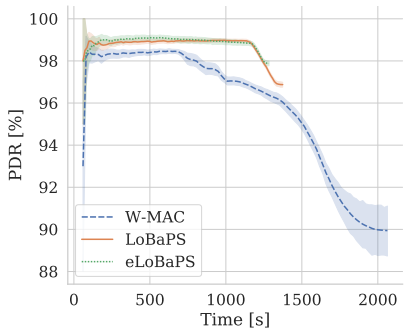

Fig. 10. Packet Delivery Ratio for IPI = 10s

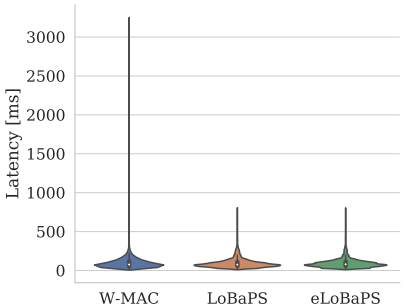

Fig. 11. Latency for IPI = 10s

6.2 Battery consumption

The battery consumption of each node at only one hop away from the sink is depicted in Fig. 8 at the instant when the first node of the network dies, for a 10 s IPI as an example of the general behavior. The topology of the network imposes constraints to the amount of load balancing that can be achieved with good performance. Leaf nodes (2 hops away from the sink) do not consume the same order of energy than relays, because they do not wake up often to listen to the MR channel. This is the reason why we do not show their battery consumption in this figure. However, the topology still generates more energy consumption on some nodes (nodes 8 and 9) because they have more chances to be woken up by some child and waste energy listening to the MR channel. The point of this figure is to analyze how equally distributed is the energy among the network. The goal is to have all nodes consuming approximately the same amount of battery when the first one dies. In the ideal case where the network consumes all the batteries in a balanced way we would expect that all the nodes die at the same time, that is, showing 100 % battery consumption. However, we can see that in W-MAC there are nodes that have only consumed half of their batteries when the first node that dies consumed it all. That amount of remaining battery not used is the reason why the lifetime is shorter because the load is not equally shared. On the contrary, in LoBaPS, the maximum remaining battery for a node at one hop from the sink is of 22%, while in eLoBaps it is of 7%, proving the improvement of the load balancing algorithm in terms of energy efficiency.

6.3 Packet Delivery Ratio

The reliability is studied in Fig. 10 by analyzing the evolution of the Packet Delivery Ratio (PDR) over time for a 10 s IPI in mean values and with the 95% confidence interval. Previous works [9] have shown the benefits of LoBaPS in contrast to W-MAC because of the stable behavior in steady state and the better final PDR. In addition, it is possible to see the decline of the PDR after

the first node dies, which happens between 750 s and 1200 s for W-MAC, and around 1200 s in LoBaPS and eLoBaPS. In W-MAC protocol, the PDR decreases fast and with high variability after this point, while in LoBaPS there is good stability during the network lifetime and a precise and controlled decline slope. eLoBaPS improves this point by reducing the length of the decline, so there is a better ratio of stable time over decline time.

6.4 End-to-end Latency

Fig. 11 shows the end-to-end latency as a violin plot that takes into account every successful packet transmission in all repetitions for an IPI of 10 s. In eLoBaPS, the latency is similar to the one achieved by LoBaPS, but in this case, it only depends on the number of retransmissions because there is no backoff exponential. The actual amount of time of the backoff period in eLoBaPS is adjusted so that it is similar to that of the average CSMA backoff period in LoBaPS. In this plot, we can also see that the lack of precision and the inefficient parent selection after a parent dies in W-MAC can lead to extremely high maximum values of latency, while in LoBaPS and eLoBaPS, the maximum is three times smaller.

6.5 Control overhead

We also consider a metric to measure the number of control packets transmitted on the whole network:

$$c = 100 \frac{\#network\ control\ packets}{\#\ app\ packets\ at\ the\ sink} \tag{5}$$

The results are shown in Fig. 12 evidencing a low control overhead for both LoBaPS and eLoBaPS compared to W-MAC. The main cause of this is that in W-MAC, whenever a backup parent dies, its children need to repair the routing structure by generating new control packets. In contrast, in both LoBaPS and eLoBaPS, this is not necessary since the initial routing structure can still be used as long as there is still connectivity in the network graph.

6.6 Productivity

Naturally, the overall application data received at the sink is different for each traffic scenario, thus making somehow doubtable the comparisons for both the PDR and the lifetime. For this, we have come up with a productivity formula that divides the number of application packets correctly received at the sink by the simulation elapsed time, so-called *simulation disconnectivity time* because the network stops being a connected graph. In addition, it is scaled by the scenario parameters (IPI and network size) so that the value can be compared between all the simulations.

$$Productivity = \frac{\#\ app\ packets\ at\ the\ sink}{disconnectivity\ time} \frac{IPI}{\#\ nodes} \tag{6}$$

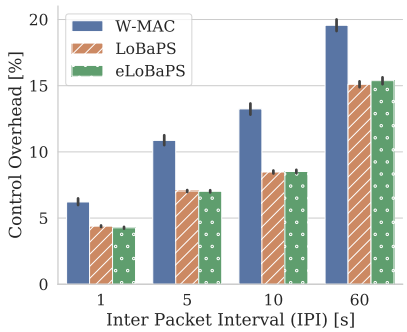

Fig. 12. Control Overhead

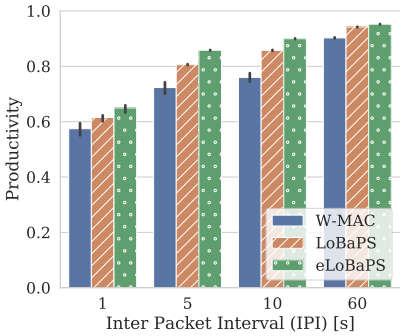

Fig. 13. Throughput for different IPIs

This way we conceive a metric of the productivity of the network that somehow combines the notion of latency, reliability and lifetime in a single value: productivity. Clearly, we want this value to be as higher as possible. We can also see this formula in an equivalent way, as the ratio of the expected disconnectivity time (calculated based on the number of application packets received at the sink, the IPI and the number of nodes) and the simulation disconnectivity time:

$$Productivity = \frac{expected\ disconnectivity\ time}{simulation\ disconnectivity\ time} \tag{7}$$

Notice that the maximum in mean values is 1 since the simulation disconnectivity time cannot be shorter than the expected one since the nodes can not generate packets faster than the IPI in mean values. So smaller values of productivity mean that it takes more time than expected to deliver the packets to the sink.

The results of this metric are illustrated in Fig. 13, where we can see a trend of improvements between W-MAC, LoBaPS, and eLoBaPS, for all the traffic loads. The reason for this is linked to the reduced number of retransmissions that are required in eLoBaPS as well as its shorter decline slope (described in Section 6.3) which turns into a smaller disconnectivity time. Couple with this result, we can see that the productivity increases as the IPI increases too for all protocols. The reason is that with low IPI there are more collisions and so fewer packets are successfully delivered to the sink.

7 Conclusions and future work

This article introduces eLoBaPS, an improvement of LoBaPS towards the ideal load balancing in wireless sensor networks exploiting the always-on feature of the wake-up radio. The main idea is to allow all feasible successors to compete for a packet forwarding when a node transmits a packet, prioritizing the ones with more remaining battery. At the same time, it mitigates the main radio listening energy wastage by turning off the most consuming nodes until they

keep up with the energy consumption of their neighbors. Moreover, the same idea can be applied with a different metric instead of the energy in order to optimize a different parameter of the network, for instance, the packet queue size. By improving the load balancing towards the ideal case, it extends the network lifetime up to 77%. In addition, the network behavior becomes more stable over its lifetime and the decline with degraded performance is shorter.

An important point that can be concluded based on the results with varying IPI, is that it is not necessary to make an effort on the design of the protocol for low traffic scenarios (high IPI). In those scenarios, the radio communication power is not significant. In contrast, research efforts should focus on high traffic scenarios (that is, with an IPI of 10 s or less as suggested by our results), thus emphasizing the importance of the CCA function already discussed in our past and present works.

Although in the long run, the parents with the best quality links probably win the competition more often than parents with bad links, the main drawback of this proposal is that nothing ensures that the most reliable end-to-end route is chosen. In our next steps, we plan to investigate this and focus on the third layer of the communication stack as well as performing real experiments in the FIT-IoT lab [2].

Acknowledgments

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

References

1. Allied Market Research https://www.alliedmarketresearch.com/smart-cities-market
2. Adjih, C.e.a.: FIT IoT-LAB: A large scale open experimental IoT testbed. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 459–464 (Dec 2015)
3. Ait Aoudia, F., Gautier, M., Berder, O.: OPWUM: Opportunistic MAC Protocol Leveraging Wake-Up Receivers in WSNs. Journal of Sensors (Jan 2016). https://doi.org/10.1155/2016/6263719, https://hal.inria.fr/hal-01244800
4. Basagni, S., Valerio, V.D., Koutsandria, G., Petrioli, C., Spenza, D.: 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). pp. 257–265 (Oct 2017)
5. Basagni, S., Valerio, V.D., Koutsandria, G., Petrioli, C.: Wake-Up Radio-Enabled Routing for Green Wireless Sensor Networks. In: 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall). pp. 1–6 (Sep 2017)
6. Magno, M., Jelicic, V., Srbinovski, B., Bilas, V., Popovici, E., Benini, L.: Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver. IEEE Trans. Ind. Informat. **12**(2), 633–644 (April 2016)
7. Piyare, R., Murphy, A.L., Kiraly, C., Tosato, P., Brunelli, D.: Ultra low power wake-up radios: A hardware and networking survey. IEEE Communications Surveys Tutorials **19**(4), 2117–2157 (Fourthquarter 2017)

8. Piyare, R., Istomin, T., Murphy, A.L.: Waco: A wake-up radio cooja extension for simulating ultra low power radios. In: EWSN (2017)
9. Sampayo, S.L., Montavont, J., Noel, T.: LoBaPS: load balancing parent selection for RPL using Wake-Up radios. In: 2019 IEEE Symposium on Computers and Communications (ISCC) (IEEE ISCC 2019). Barcelona, Spain (Jun 2019)
10. Shukla, S.N., Champaneria, T.A.: Survey of various data collection ways for smart transportation domain of smart city. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). pp. 681–685 (Feb 2017)
11. Thubert, P.: Objective function zero for the routing protocol for low-power and lossy networks (rpl). RFC 6552, RFC Editor (March 2012)