LoRa-DuCy: Duty Cycling for LoRa-Enabled Internet of Things Devices

1st Thenuka Karunathilake
University of Bremen
Bremen, Germany
thenuka@comnets.uni-bremen.de

2nd Asanga Udugama
University of Bremen
Bremen, Germany
adu@comnets.uni-bremen.de

3rd Anna Förster
University of Bremen
Bremen, Germany
anna.foerster@uni-bremen.de

***Abstract*—The LoRa (Long Range) communication technology has gained a lot of interest recently. Typically, Internet of Things applications would use the complete LoRaWAN stack for their purposes. However, LoRaWAN supports only three types of communication, called classes A, B, and C. Each of these classes addresses a different application scenario and resource usage goals, but all of them target star topologies, where devices communicate only to the gateway. On the other hand, wireless sensor networks (WSNs) have been successfully developing and using the concept of self-organised networks and duty cycling. In this paper, we combine these fields and implement a WSN duty-cycling medium access protocol, known as Contiki-MAC, on LoRa-enabled LoPy4 devices. We experimentally evaluate our implementation and we compare its power consumption with that of LoRaWAN-enabled devices. We show that our implementation is more power-efficient while achieving higher traffic rates and offering the freedom of self-organised networking for various applications.**

I. Introduction

The exponential growth of IoT (Internet of Things) in the past decade has spread its applications to every field of everyday life from industrial applications to art and culture. Various communication technologies can be used to interconnect IoT devices, but LoRa (Long Range) has experienced a growing interest. The main advantages of LoRa are the extended wireless range and low power consumption. Normally, IoT devices are deployed without a continuous power supply and most of them are battery-powered or equipped with an energy harvesting method. In either case, the power consumption is an important factor to consider because it determines the node's lifetime and thus also the costs of the system. In an IoT device, the radio transceiver is among the highest power-consuming components, and optimising its power consumption will significantly improve the lifetime.

Duty cycling has been traditionally used as the main driver for saving energy in wireless sensor networks (WSN) [16]. The current LoRaWAN protocol stack already employs duty cycling to save energy by introducing three different classes (Class A,B, and C). However, the main drawback in LoRaWAN architecture is that all end-nodes have to communicate directly with a LoRa gateway. Therefore, such a network with multiple gateways is more expensive and furthermore, generates safety vulnerabilities at the gateway and those aspects also have to be considered while deploying [23].

To reduce the number of LoRa gateways, a viable solution is to apply a multi-hop communication where LoRa nodes are used to route information in an AdHoc manner. The importance of such a multi-hop communication is emphasized by authors in [19]. In such a communication network, to avoid collisions and to increase channel utilization, a Medium Access Control (MAC) protocol is a must to have. Furthermore, an efficient MAC protocol can provide much-needed radio duty cycling to reduce idle listening resulting in better consumption in battery-powered devices.

Therefore, in this paper, we propose to implement one well-known WSN MAC protocol, namely Contiki-MAC [5] over LoRa physical layer. Our contributions are:

- Implementation of Contiki-MAC for LoRa devices
- An experimental performance evaluation
- Energy consumption comparison between the new implementation and LoRaWAN

The rest of the paper is organised as follows. Section II discusses the state of the art WSN MAC protocols and LoRa related works. Section III provides background on Contiki-MAC. Section IV presents our performance evaluation on LoPy devices, while Section V focuses on the power consumption comparison against LoRaWAN. Finally, Section VI concludes the paper.

II. Related Work

There exist plenty of MAC protocols for WSNs. In general, they are all based on duty cycling the radio and can be divided into synchronous and asynchronous. Synchronous protocols include S-MAC [22], T-MAC [20] and TSMP [15]. However, they require time synchronisation, which is not always available or incurs high costs.

In asynchronous methods, there is no synchronisation between nodes, which makes them less complex. Further, asynchronous protocols can be divided into sender-initiated and receiver-initiated protocols [5]. In sender-initiated protocols, nodes wake-up periodically and sense the channel for ongoing transmission. Most commonly used examples are B-MAC [16], X-MAC [4], BoX-MAC [12], WiseMAC [7] and Contiki-MAC [5] [11]. In receiver-initiated protocols, the responsibility of initiating the transmission is given to the receiver. Some examples are Koala [13], RI-MAC [17] and A-MAC [6].

However, sender-initiated protocols are more popular and have been widely used in the community. Therefore, we selected Contiki-MAC for our implementation because of following reasons. Apart from traditional benefits in asynchronous protocol such as less complexity and low overhead, Contiki-MAC consists of one of the most efficient wake up mechanisms in the radio duty cycling family. Furthermore, Contiki-MAC in-cooperates most efficient techniques introduced in earlier protocols (phase lock optimisation from WiseMAC [7] and replacing wake up strobe with multiple data copies from BoX-MAC [12]).

During our implementation, we selected LoRa communication because it has become an ideal candidate for many IoT applications mainly due to the long-range and low power features. One such application is wildlife monitoring and in [21] authors have compared several low-cost radio transceivers and found out that LoRa was able to outperform other transceivers.

As mentioned in Section I, the current LoRaWAN stack is equipped with duty cycling and it is restricted to 1% by the standard. Therefore, in some applications, the occasional transmission of a large amount of data is not possible. As a solution, [3] proposes an activity time-sharing mechanism to allow a node to use additional activity time of other nodes. The limitation of scalability in current ALOHA based LoRaWAN MAC is explored in [10] and a solution for this drawback is provided in [24] with time-slotted communication. Additionally, to overcome the collisions in ALOHA based schemes, several studies have explored the Carrier Sense Multiple Access (CSMA) technique in LoRa networks [18] [8]. Similar work is carried out in [2] to avoid a higher number of collisions in ALOHA based LoRaWAN with a modified listen-before-talk (LBT) mechanism. The discussed approaches do not provide a full solution and most of them are focused on LoRaWAN and thus, we propose to use a WSN MAC protocol such as Contiki-MAC over LoRa physical layer.

In a MAC protocol, avoiding collisions is among the highest priorities. To achieve this, nodes must be able to perform carrier sensing to detect channel activity. LoRa modules can use two different techniques for carrier sensing [14]. First, with Receive Signal Strength Indicator (RSSI) the received RF energy at the antenna can be used to measure the channel occupancy. The second method is called Channel Activity Detection (CAD) which is a LoRa specific feature. It is designed to identify LoRa preamble signals and the main advantage of this mechanism is being able to work below the noise floor. A key drawback of this mechanism is not being able to detect the payload part of the signal [14]. Therefore, the payload can still cause collisions. However, the most recent LoRa modules (SX126x) claim to handle the payload detection with CAD. During our implementation we used SX127x and therefore, we opted to select the first mechanism as our carrier sensing method.

III. Contiki-MAC

Contiki-MAC is a successor and a combination of various other WSN MAC protocols, such as B-MAC [16], X-MAC [4], BoX-MAC [12], and WiseMAC [7], and additionally introducing some optimisations to further save energy. It is part of the ContikiOS for embedded devices[1] and it is one of the best explored and mostly used MAC protocols for WSNs. An overview of the Contiki-MAC protocol is depicted in Figure 1, based on [5]. This Section briefly discuss the Contiki-MAC mechanism and detailed descriptions are available in [5] [11].

The driving parameter of any duty cycling MAC protocol is the wake-up interval. At regular intervals, Contiki-MAC wakes up and senses the channel for activity (clear channel assessment - CCA). In case the CCA detected an activity, it starts receiving the packet and if no activity is detected, Contiki-MAC goes into a transmit mode and differentiates between unicast and broadcast transmissions.

There are several important optimisations in Contiki-MAC namely fast sleep, phase lock loop (PLL), and broadcast optimisation. The latter two optimizations are depicted in Figure 1 with red colour.

Fast sleep optimisation is used to identify the false positives from CCA and prevents long awake periods in case of radio noise. PLL optimisation is used to reduce the unnecessary transmissions to the same neighbour during unicast transmission by using previous acknowledgements from that particular neighbour. Finally, broadcast optimisation suggests switching off the radio between two repetitions of the packet (Packet interval) to save power because in broadcast, acknowledgements are not expected.

IV. Experiments and Performance Analysis

We have implemented Contiki-MAC on pycom LoPy4 devices [1]. These devices are popular among the IoT developing community as they are equipped with multiple communication technologies (LoRa, WiFi, Bluetooth, and Sigfox), easy to program with MicroPython and well documented. We call our implementation LoRa-DuCy (LoRa on Duty Cycles) and have made it publicly available under GitHub[2].

The used Contiki-MAC parameter values with LoRa are presented in Table I with original Contiki-MAC work parameters as described in [5]. The LoRa parameters were set to spreading factor 7 and coding rate 4/5. We selected a packet size of 255 bytes because it is the maximum allowed LoRa payload. Since this 255 bytes packet is much larger than the 23 bytes used in the original work and additionally, to cater to slow data rates in LoRa, all other parameters (CCA duration, CCA interval, and Packet interval) had to be adjusted from the original values. These timing parameters are selected and set up after initial experiments to make sure that communication is mostly successful.

The conducted experiments were focused on two main areas: a performance analysis to show the overall effectiveness of LoRa-DuCy and a power consumption analysis (Section V) to compare the power consumption of the LoRa-DuCy against LoRaWAN.

[1] http://www.contiki-os.org/

[2] https://github.com/ComNets-Bremen/LoRa-DuCy

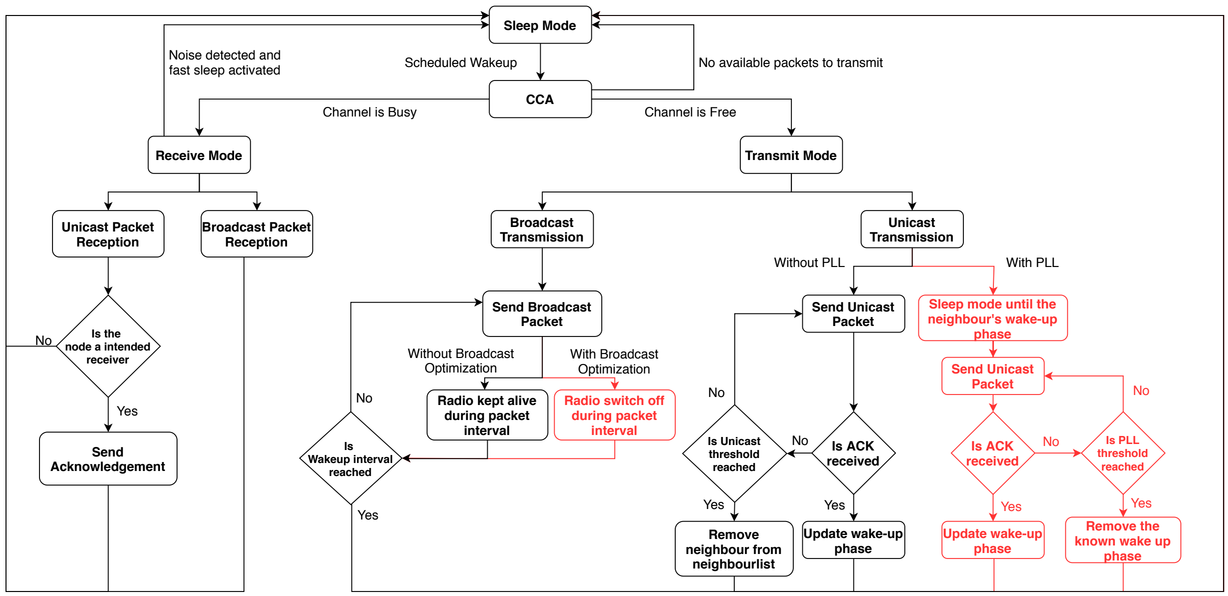

Fig. 1. Contiki-MAC mechanism as described in [5]

TABLE I
CONTIKI-MAC PARAMETER VALUES USED DURING THE IMPLEMENTATION

Parameter	Values with LoRa	Original work values as described in [5]
Wakeup interval	5 – 40 sec	30 - 200 ms
Packet size	255 bytes	23 bytes
CCA duration	80 ms	0.192 ms
CCA interval	400 ms	0.5 ms
RSSI threshold	100 dBm	N.A
Packet interval	500 ms	0.4 ms
ACK packet size	15 bytes	N.A

Delivery ratio and duty cycle values were used as metrics during the performance analysis. The delivery ratio was calculated as how many packets were successfully delivered from all the transmitted packets. The duty cycle value meant how much time the node's radio was kept in active mode during the full experiment period, which depends not only on the parameter values such as wakeup interval but also on the actual traffic and the transmission success. All results have been presented as averages from several experiments with confidence intervals computed at a 95% confidence level.

We conducted experiments with 10 LoPy4 devices. The nodes contained all possible expansion board options for the LoPy4 device (Pysense, Pytrack, and Expansion board 3.0) and during our tabletop experiments all the LoPy4 devices were connected to a laptop. The experiment duration was set to 1 hour. During this time, each node was configured to generate packets randomly.

A. Traffic impact

First, we evaluate the impact of different types of traffic on the performance of LoRa-DuCy. For this, we use three scenarios and we activated the respective optimisations:

Scenario 1: Nodes do not transmit any data
Scenario 2: Nodes transmit broadcast data
Scenario 3: Nodes transmit unicast data

The obtained duty cycle values for the three scenarios are depicted in Figure 2(a). As expected the lowest duty cycles were achieved in scenario 1 because in this configuration nodes do not transmit any data. The highest duty cycle values were obtained in scenario 2 because in broadcast mode the transmission must continue for the full wake-up interval. In contrast, in unicast mode nodes only transmit until the acknowledgement. Therefore, scenario 3 shows lower duty cycle values compared to scenario 2.

In scenario 2 and scenario 3 LoRa-DuCy reacted similarly to higher traffic amount by increasing the overall duty cycle. This clearly implies that the node needs to spend more time in active mode to transmit more data packets. However, the increment in duty cycle values was significantly different in unicast mode and broadcast mode because of their separate active optimisations.

The obtained delivery ratios are illustrated in Figure 2(b) for the broadcast scenarios and Figure 2(c) for unicast scenarios. The wake-up intervals above 20 seconds were able to achieve more than 90% delivery rates. However, smaller wake-up intervals below 20 seconds depicted relatively lower delivery rates due to overlapping CCA.

B. Impact of phase lock loop optimisation

Here, we evaluate the impact of the PLL optimisation on the performance of LoRa-DuCy. For this, we use the unicast scenario (scenario 3) from above, since PLL only influences unicast transmissions. For better comparison, we also plot the no transmissions scenario (scenario 1).

The results are provided in Figure 3 for different traffic volumes. The dotted lines represent the duty cycle values

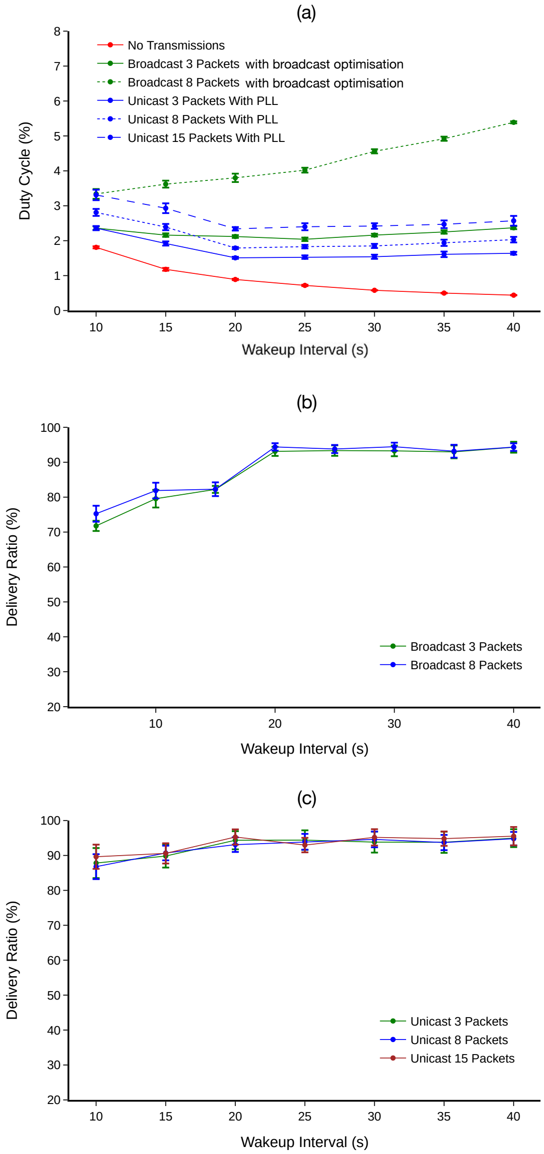

Fig. 2. Obtained duty cycles and delivery rates with different traffic volumes.

without the PLL optimisation and corresponding solid lines indicate the duty cycle values with the PLL optimisation. The PLL was able to perform significantly better in every wake-up interval compared to non optimised versions. The achieved benefit is increasing with longer wake-up intervals because PLL can permit a node to stay longer in the sleep mode. This behaviour is illustrated in Figure 3 with an increasing gap between the dotted line and the solid line as the wake-up interval increases. Additionally, with PLL the cost of delivering a packet is similar in every wake-up interval. Therefore, the gradient of the solid line was significantly decreased. *It can be summarised that PLL is a very effective measure to minimise the duty cycles of LoRa-DuCy.*

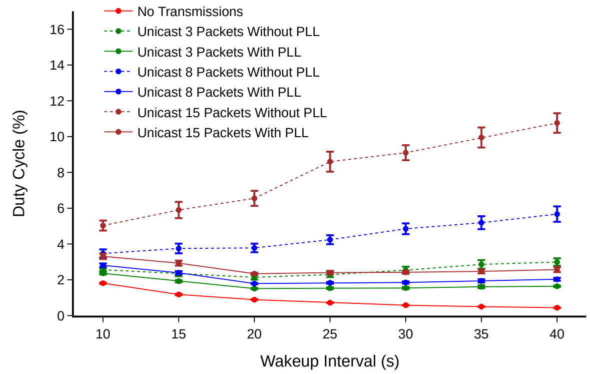

Fig. 3. Impact of PLL optimisation with different traffic volumes.

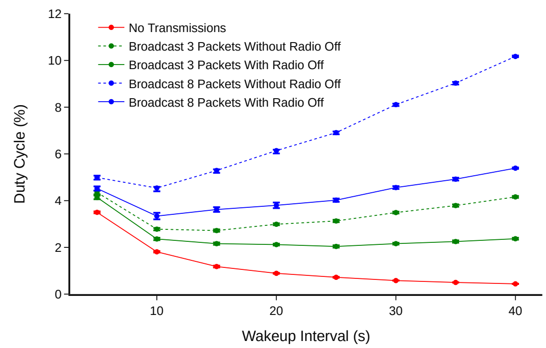

Fig. 4. Impact of broadcast optimisation with different traffic volumes.

C. Impact of broadcast optimisation

Figure 4 shows the impact of the broadcast optimisation and the dotted lines represents the duty cycle values without the optimisation and the solid line depicts the performance with the broadcast optimisation.

In broadcast, with shorter wake-up intervals the time spent on transmitting is shorter than for longer wake-up intervals. Hence, the resulting duty cycle values were increased with longer wake-up intervals. A significant improvement can be achieved by the broadcast optimisation and the obtained benefit depends on the length of the wake-up interval. Because in longer wake-up intervals, broadcast optimisation is used more frequently due to a higher number of transmissions. Furthermore, similar to unicast mode all the optimised and non optimised duty cycle values were increased with higher traffic amounts. *Thus, also the broadcast optimisation has a significant positive effect on the performance of the protocol.*

V. Power Consumption Analysis

We conducted power measurements with an oscilloscope and a power supply. During all the power measuring experiments, a LoPy4 device was used with a pysense expansion

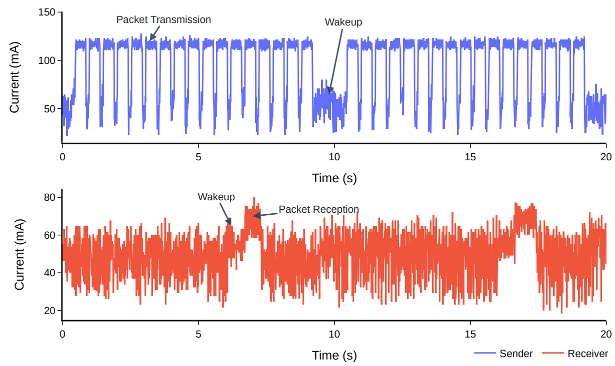

Fig. 5. LoRa-DuCy broadcast current consumption.

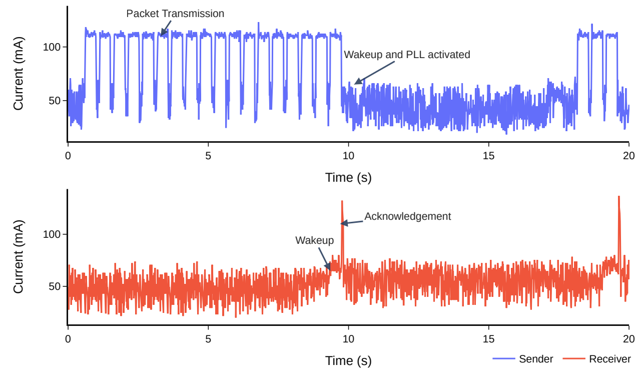

Fig. 6. LoRa-DuCy unicast current consumption.

TABLE II
CURRENT CONSUMPTION VALUES FOR DIFFERENT STATES WITH LORAWAN, LORA-DUCY AND WITHOUT MAC

Mode	Transmit State	Receive State	Sleep State
LoRa-DuCy	115 mA	70 mA	52mA
Without MAC	120 mA	68 mA	-
LoRaWAN Class A	115 mA	68 mA	57mA
LoRaWAN Class C	115 mA	68 mA	-

board. The device was connected in series with a known resistor and the voltage drop across the resistor was used to calculate the current in the network. The power consumption analysis was conducted in three different modes: with LoRa-DuCy, LoRaWAN, and without any MAC protocol.

A. Power Consumption of LoRa-DuCy

First, the node was configured to operate in broadcast mode and the wake-up interval was set to 10 seconds, while a second node was a receiver. The obtained power measurements are depicted in Figure 5. In the broadcast mode, the sender continuously transmitted multiple copies during the full wake-up interval and this behaviour is clearly visible in the figure. In the transmit mode the device used a current of around 115 mA. As depicted, the receiver wakes up at the scheduled time, performed the CCA and was able to identify the ongoing transmission. The receive state consumed a current of around 70 mA and after packet reception, the node returned to sleep state which consumed around 52 mA

Next, we configured the device to work in the unicast mode with 10 seconds wake-up interval. As illustrated in Figure 6, the first transmission was done normally without the PLL optimisation and the neighbour's wake-up phase was identified by the acknowledgement. After receiving the data packet, the receiver transmitted an acknowledgement packet and this is depicted in the figure with a short peak after the receive state. The following transmission to the same receiver was now optimized by the PLL and the transmission only begins just before the neighbour's scheduled wake up. Until that time sender remained in the sleep state and consumed around 52 mA. Similar to the broadcast mode, the unicast transmit state and the receive state consumed around 115 mA and 70 mA respectively.

B. Power Consumption Comparison

Next, we compared the power consumption of LoRa-DuCy with LoRaWAN mode and a "without MAC" mode. For LoRaWAN we have configured the nodes to work in class A or class C configurations. While LoRaWAN always employs a gateway, class A devices are configured asynchronously and can go back to sleep at any time. Class C devices are always on, while class B devices operate with periodic beacons from the gateway [9]. However, class B configuration is not available for our Pycom devices. In the mode "without MAC", the node is always on and blindly transmits packets to other nodes without sensing the channel for any ongoing transmissions. Therefore, collisions are expected and retransmissions are not possible.

All measured power consumption values are summarised in Table II. LoRaWAN class C and the mode "without MAC" consume the most power because they operate without a sleep state. LoRa-DuCy and the LoRaWAN class A were the most power-efficient modes because of the sleep state. However, during the sleep state LoRaWAN class A consumed slightly higher current than LoRa-DuCy.

Finally, we conducted an experiment with 5 LoPy4 devices. Each device was equipped with a pysense expansion board and a 1000 mAH battery. In each configuration, a packet transmission was scheduled every 120 seconds. The LoRa-DuCy mode was set to operate with 20 seconds wake-up interval. The overall aim of the experiment was to find out how much time can a node survive with the available battery capacity. Each mode was tested in five experiments and the obtained results are illustrated in Figure 7 (averages with confidence intervals at 95% confidence level).

As expected the mode without the MAC protocol and the LoRaWAN class C configurations depicted the shortest life-

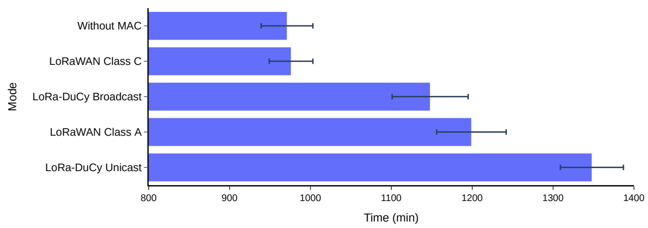

Fig. 7. Average number of minutes the nodes lasted with 1000 mAH battery for LoRaWAN, LoRa-DuCy and without MAC

times because these modes operate without a sleep state. The LoRa-DuCy broadcast was the next most power-hungry mode because in broadcasting same packet must be transmitted with multiple copies. As explained in earlier sections the unicast was able to out-perform broadcast because PLL optimisation was able to reduce the required number of transmissions. The LoRaWAN class A depicted a slightly shorter lifetime than LoRa-DuCy unicast mode. The reason for this shorter lifetime in class A was slightly higher power consumption in sleep state as shown in Table II.

VI. Summary and Conclusions

In this paper, we presented our implementation of ContikiMAC over the LoRa physical layer on LoPy devices, called LoRa-DuCy. We have not only shown that the implementation is feasible, but we also validated it with real-world experiments and have explored the impact of its various optimisations on the performance with different traffic volumes.

From our conducted experiments, we showed that the implemented LoRa-DuCy was able to achieve more than 90% delivery ratio. Additionally, we measured the power consumption of LoPy4 devices working with LoRa-Ducy in different modes. Furthermore, we have compared its power consumption to LoRaWAN and have demonstrated its potential to save energy. With this implementation, which is also available open-source, the research community can achieve higher traffic volumes and enable multi-hop communications. We plan to use this implementation in our ongoing work in underground sensor networks [23].

References

[1] Pycom - Next Generation Internet of Things Platform. https://docs.pycom.io/. Library Catalog: pycom.io.

[2] S. Ahsan, S. A. Hassan, A. Adeel, and H. K. Qureshi. Improving Channel Utilization of LoRaWAN by using Novel Channel Access Mechanism. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1656–1661, June 2019. ISSN: 2376-6506.

[3] N. Benkahla, H. Tounsi, Y.-Q. Song, and M. Frikha. Enhanced Dynamic Duty Cycle in LoRaWAN Network. In N. Montavont and G. Z. Papadopoulos, editors, *Ad-hoc, Mobile, and Wireless Networks*, Lecture Notes in Computer Science, pages 147–162, Cham, 2018. Springer International Publishing.

[4] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. volume 4, pages 307–320, Jan. 2006.

[5] A. Dunkels. The ContikiMAC radio duty cycling protocol. May 2012.

[6] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 1–14, Zürich, Switzerland, Nov. 2010. Association for Computing Machinery.

[7] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. In S. E. Nikoletseas and J. D. P. Rolim, editors, *Algorithmic Aspects of Wireless Sensor Networks*, Lecture Notes in Computer Science, pages 18–31, Berlin, Heidelberg, 2004. Springer.

[8] M. O. Farooq and D. Pesch. A Search into a Suitable Channel Access Control Protocol for LoRa-Based Networks. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 283–286, Oct. 2018. ISSN: 0742-1303.

[9] C. E. Fehri, M. Kassab, S. Abdellatif, P. Berthou, and A. Belghith. LoRa technology MAC layer operations and Research issues. *Procedia Computer Science*, 130:1096–1101, 2018.

[10] A. Lavric and V. Popa. Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks, June 2018. ISSN: 1530-8669 Pages: e6730719 Publisher: Hindawi Volume: 2018.

[11] P. J. Marrón and K. Whitehouse. *Wireless Sensor Networks: 8th European Conference, EWSN 2011, Bonn, Germany, February 23-25, 2011, Proceedings*. Springer Science & Business Media, Feb. 2011. Google-Books-ID: D3OR7uCztGwC.

[12] D. Moss and P. Levis. BoX-MACs: Exploiting Physical and Link Layer Boundaries in LowPower Networking. Technical report, 2008.

[13] R. Musaloiu-E., C.-J. M. Liang, and A. Terzis. Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 421–432, Apr. 2008.

[14] M. O'Kennedy, T. Niesler, R. Wolhuter, and N. Mitton. Practical evaluation of carrier sensing for a LoRa wildlife monitoring network. page 8.

[15] K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. *IASTED International Symposium on Distributed Sensor Networks, DSN 2008*, Jan. 2008.

[16] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, page 95, Baltimore, MD, USA, 2004. ACM Press.

[17] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, page 1, Raleigh, NC, USA, 2008. ACM Press.

[18] T. To and A. Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018. ISSN: 1938-1883.

[19] V. Toldov, L. Clavier, and N. Mitton. Multi-channel Distributed MAC protocol for WSN-based wildlife monitoring. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, Oct. 2018. ISSN: 2160-4886.

[20] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, Los Angeles, California, USA, Nov. 2003. Association for Computing Machinery.

[21] J. Wotherspoon, R. Wolhuter, and T. Niesler. Choosing an integrated radio-frequency module for a wildlife monitoring wireless sensor network. In *2017 IEEE AFRICON*, pages 314–319, Sept. 2017. ISSN: 2153-0033.

[22] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1567–1576 vol.3, June 2002. ISSN: 0743-166X.

[23] I. Zaman and A. Förster. Challenges and opportunities of wireless underground sensor networks. In *Fachgespräche sensornetze*, 2018.

[24] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch. TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things. *Computer Communications*, 153:1–10, Mar. 2020.