Enhanced ADR for LoRaWAN networks with mobility

Norhane Benkahla, Hajer Tounsi, Yeqiong Song, Mounir Frikha

To cite this version:

Norhane Benkahla, Hajer Tounsi, Yeqiong Song, Mounir Frikha. Enhanced ADR for LoRaWAN networks with mobility. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, Jun 2019, Tanger, Morocco. pp.1-6, 10.1109/IWCMC.2019.8766738. hal-02975202

HAL Id: hal-02975202

https://hal.archives-ouvertes.fr/hal-02975202

Submitted on 22 Oct 2020

Enhanced ADR for LoRaWAN networks with mobility

Norhane BENKAHLA
MEDIATRON, SUP'COM
Carthage university
Tunis, Tunisia
norhane.benkahla@supcom.tn

Hajer TOUNSI
MEDIATRON, SUP'COM
Carthage university
Tunis, Tunisia
hajer.tounsi@supcom.tn

Ye-Qiong SONG
LORIA
University of Lorraine
Nancy, France
ye-qiong.song@loria.fr

Mounir FRIKHA
MEDIATRON, SUP'COM
Carthage university
Tunis, Tunisia
m.frikha@supcom.tn

***Abstract*—LoRa is becoming an attractive low cost and low power WAN solution for many real-world IoT applications. LoRa has been designed for static end-devices to individually use the optimal configuration through an adaptive data rate mechanism (ADR), thanks to the possibility to choose a set of LoRa physical layer transmission parameters. However a large class of IoT applications (e.g. connected farm) also includes mobile nodes with specific mobility patterns. For those applications, the current ADR control algorithm may not be efficient when the radio channel attenuation rapidly changes because of the node mobility. This paper contributes to enhance the ADR mechanism by taking into account the position of the mobile devices and their trajectories in order to have a dynamic allocation. The Enhanced-ADR (E-ADR) minimizes the transmission time and energy consumption as well as packet loss for mobile devices. The testbed-based experiments show that E-ADR improves the quality of service (QoS) of the overall networks.**

***Index Terms*—IoT, LoRaWAN, ADR, Spreading factor, Positioning, mobility**

I. INTRODUCTION

The Internet of Things (IoT) community is currently focusing on the design of large-scale network infrastructures targeting the coverage of massive-scale city-wide scenarios using LPWAN technologies such as LoRaWAN. The LoRaWANs architecture is built on a star topology where multiple LoRa End-Devices (ED) are interconnected to one or many Gateways (GW). A message transmitted by an end-node can be received by multiple close and far-away gateways, which in turn forward the collected messages to a Network Server.

Communication between end-devices and gateways is spread out on different frequency channels and data rates. The selection of the data rate is a trade-off between communication range and message transmission time [1]. Communications with different data rates do not interfere with each other thanks to the choice of different spreading factors and frequency channels. The LoRa network server can manage the data rate and the transmission power of each device individually to maximize both the battery life and the overall network capacity by means of the Adaptive Data Rate (ADR) scheme which allows to choose an optimal combination (mode) among a range of parameters: Spreading factor (SF), Bandwidth (BW), Code Rate (CR), Transmission power (TP), Data Rate (DR). Accordingly, there is a trade-off between SF and communication range. The higher the SF, the slower the transmission and the longer the communication range [2].

The main challenge in this case is to determine the right configuration that allows a reliable communication with a low energy consumption.

However, the basic ADR scheme may not be efficient in the case of mobile nodes. Indeed the current rate adaptation is performed only after the reception of a set of frames (20 frames in default ADR) [1]. It does not always allow to take into account the degradation of the signal due to either the node mobility or the presence of moving obstacles. Our experiments revealed two issues: waiting for collecting 20 frames may be too long for quickly adapting to new situations and transmitting 20 frames may be too many, leading some times to increasing the Time on Air (ToA) beyond authorized 1% duty-cycle imposed by the regulation authorities [1].

In this paper, we propose E-ADR, an enhanced ADR scheme aiming at self-adapting to the node mobility through quickly reconfiguration of the mode. E-ADR is based on estimating the next position of a mobile device and its predefined trajectory. Thus the network server could define the best configuration mode for the device to reduce power consumption, channel occupancy time and packet loss rate. E-ADR has been implemented using Waspmote SX1272. It has been tested and compared to the basic ADR in a connected farm scenario.

The rest of the paper is organized as follows. In Section 2, we present the LoRa/LoRaWAN technology and the ADR strategy in LoRaWAN networks. Section 3 highlights related works. Section 4 describes E-ADR in detail. Our implementation and our experimentation scenario will be discussed in Section 5 by showing how the proposal is adapted to the mobile and fixed devices. We conclude in Section 6.

II. LORA/LORAWAN

A. *LoRa/LoRaWAN background*

A LoRa network is based on two components, namely LoRa and LoRaWAN. LoRa is a proprietary physical layer developed by Semtech's Corporation. LoRaWAN (medium access control protocol) is described in an open specification developed by the LoRa Alliance [3]. The LoRa physical layer uses Chirp-Spread Spectrum (CSS) modulation to enable long distance and low power communications. LoRa operates in the sub-GHz ISM band. Each LoRa transmission is characterized by five parameters: Spreading factor, transmission power, code rate, data rate and bandwidth [4]. These settings affect communication range, data rate, robustness to interference or noise and the ability of a receiver to decode the signal. The configuration of spreading factors allows us to adjust the data rate and the accessible distance. TABLE I represents the different configuration modes used in Lora Waspmote devices [4] [5], where each mode n corresponds to a range of RSSI (Received Signal Strength) bounded by $[B_{inf}(n),\ B_{sup}(n)]$ (except mode 10).

TABLE I: Configuration modes [4]

mode	**BW (kHz)**	**CR**	**SF**	**RSSI (dB)**	**DR (Kbps)**
1	125	4/5	12	$[-134, -131]$	0,293
2	250	4/5	12	$[-131, -129]$	0,585
3	125	4/5	10	$[-129, -128]$	0,976
4	500	4/5	12	$[-128, -126]$	1,718
5	250	4/5	10	$[-126, -125.5]$	1,953
6	500	4/5	11	$[-125.5, -123]$	2,148
7	250	4/5	9	$[-123, -120]$	3,515
8	500	4/5	9	$[-120, -117]$	7,031
9	500	4/5	8	$[-117, -114]$	12,50
10	500	4/5	7	$[\geq -114]$	21,875

To maximize both the battery life of the end-devices and overall network capacity, LoRa can manage the data rate and transmission power for each end-device individually by means of an adaptive data rate (ADR) scheme [3]. A device interested in adapting its data rate activates the ADR flag in any up-link MAC frame header. When this is enabled, the network will be optimized to use the fastest possible data rate (to reduce ToA, energy consumption and increase the QoS). When the Network Server is unable to control the data rate of a device due to fast changes, the device's application layer should control it. This latter tries to minimize the aggregated ToA used given the network conditions [1].

B. Adaptive data rate scheme

The ADR mechanism in LoRa runs in two parts: on the LoRa node and on the network server. The server provides the most complex part to keep the nodes as simple as possible. The objective of the part running on the node is only to decrease the data rate for increasing the radio coverage if the up-link transmission does not reach the gateway (loss of connection). A counter defining the number of frames sent in up-link without being acknowledged is triggered. If this counter reaches a certain limit, then the node increases the SF (decreases the configuration mode) which could increase the probability of reaching the gateway. The part executed on the server makes it possible to change the transmission power and increase the data rate for the up-links. The server collects the SNRs (Signal-to-noise ratio) of the 20 frames received after activation of the ADR flag and estimates on this basis the new parameters for the future transmissions using equations (1) and (2) until the next ADR activation.

$$SNR_{margin} = SNR_{max} - SNR_{Req} - 10 \tag{1}$$

$$N_{step} = floor(SNR_{margin}/3) \tag{2}$$

In (1), SNR_{max} is the maximum SNR of the received frames, SNR_{Req} is the corresponding SNR of the last received packet's mode and 10 refers to a margin constant [1]. In (2), N_{step} is the adjustment step. In the case it is greater than 0 and the minimum SF (SF_{min}) is not yet reached, the server increases the data rate. If the SF is equal to SF_{min}, the server decreases the Transmission Power. In the case where N_{step} is less than 0 and the max power is not reached, the server increases the power otherwise it does not make change [6].

III. RELATED WORKS

Recent research on LoRa / LoRaWAN was focused on assessing LoRa performance in terms of capacity, life duration and coverage. These studies were carried out using real experiments in [7], mathematical models in [8] or simulations in [9]. Among works done for the performance evaluation, [10] shows the exponential dependence of ToA on the SF. [11] confirms also that the throughput is limited either by the collision rate or by the duty-cycle limitation. As a solution, the authors in [12] implement a dynamic transmission scheme, i.e. ADR in LoRaWAN, and densify the infrastructure by adding additional gateways.

In addition, the authors in [12] were interested in the LoRa transmission parameters. In fact, LoRa device can be configured to use different spreading factors, bandwidth settings, coding rates and transmission powers, resulting in overall 6720 possible settings. The authors in [12] have developed a link probing mechanism with different settings to determine the optimal configuration for each device based on transmission energy. However, due to its high complexity, the trade-off to be found for this solution is the reduction of the number of probes to determine the optimal parameters.

In [13], the authors considered a distribution of spreading factors based on the median of the SNR values received at the gateway instead of the maximal SNR. However, the case of obstacles and mobility is not considered and waiting for collecting 20 frames may be too long for quickly adapting to these cases.

In [14], the authors proposed two approaches to allocate SF, EXP-SF (Extending the performance using SF) and EXP-AT (Extending the performance using Air time). The goal is to achieve a high overall throughput. The principle of EXP-SF is to divide the nodes into 6 groups and assign each group a different configuration among the predefined SFs (SF7 to SF12). The first group corresponding to the nodes having the highest RSSIs are assigned the SF 7, the second group with the following RSSIs is assigned the SF8, and so on. The procedure is repeated until all subsets are served. The second approach EXP-AT is more dynamic than the first, where the allocation of the SF theoretically equalizes the transmission time of the nodes. Both approaches are interesting and have been integrated in the processing of the server in the case of the ADR scheme is activated. But in the case of EXP-SF, we can have use cases where the attributed SF is not consistent with the limitations provided in LoRaWAN which degrades performance.

In [15], authors proposed a new ADR algorithm for LoRa networks at the nodes. The principle of this algorithm is to determine the level of congestion. It performs a learning method using the different transmission parameters. This learning method uses a logistic regression algorithm. In the case where congestion is predicted, the scheme adjusts latency instead of reducing throughput. The disadvantage is that the algorithm requires an active acknowledgment for every transmission. However, this mechanism would decrease the delivery ratio as down-link traffic has been demonstrated to have an impact on up-link throughput. In summary the above reviewed works aim at improving the performance by proposing either enhancements or new ADR, however, they have not considered the mobility case and we think that the degree of their reactivity compared to frequent mobility situations is very low.

IV. CONTRIBUTION

Our goal is to find the optimal configuration of parameters that will respond to the device's needs in terms of QoS even with node mobility. The choice of a configuration has a direct impact on the range of the transmission, the ToA and the energy consumption. Thus, we propose a configuration allocation model driven by the network server. Contrary to the basic ADR, the server can either increase or decrease the configuration mode. This model is suitable for both fixed nodes and the mobile ones that will be localized to adjust the configuration they will use. The main idea of our mechanism is to define a new configuration mode according to the predicted position of a device based on its previous ones and the trilateration method. [16]. Once the new position is defined, the server calculates the corresponding RSSI and looks for the best RSSI interval in which it can be located to determine the most suitable new mode.

A. Position estimation

There are several localization techniques. [17] gives an overview of the localization techniques for wireless sensor networks. The authors have shown that the most accurate solution is the use of GPS. However, its disadvantage is the additional cost in terms of energy consumption. [18] has presented a GPS-free solution using an accelerometer and a compass. The disadvantage of this solution is that the error is cumulative over time. Other approaches that are low-cost and more energy efficient are based on the RSSI. [19] used trilateration approach that consists to define the device's position based on the information collected from three gateways. Each gateway defines the distance that separates it from the device based on the RSSI of the received packet. The distance between a device and a gateway represents the radium of the circle having as center the gateway's position. The intersection of the three circles of the gateways represents the adjusted position of the device. [20] has used angle of arrival (AoA) technique where the location of a node is estimated by 2 base stations equipped with antenna array. The Time Difference Of Arrival (TDOA) technique [21] has used three or more gateways with precise time references.

In our work, we choose the trilateration technique for its very low complexity. We assume that the network server knows the positions of the gateways. To deduce the distance separating the node from the gateway from the measured RSSI, we adopted a tendency curve presented in Fig. 1 that we have obtained experimentally. We draw the curve by moving our device from the gateway by a distance of 1 m for each test and measuring the RSSIs of 100 packets sent. An average RSSI is taken with a 95% confidence interval. We note that this calibration phase should be conducted every time we have a new deployment area. How to make it more general is part of our future work.

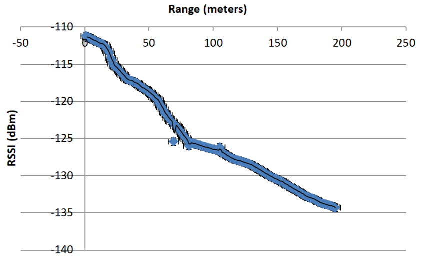


Fig. 1: LoRa range measurement vs. RSSI

To estimate the next position of a mobile device, we calculate an average displacement based on the last n measured positions $P_i = (X_i, Y_i)$ according to equation (3).

$$Avg_{displacement}(X, Y) = \frac{\sum_{i=0}^{n-1}(P_{i+1} - P_i)}{n-1} \tag{3}$$

The position of the device sending the $(i+1)^{th}$ packet is defined as $P_{i+1}(X_{i+1}, Y_{i+1})$ is equal to $(X_i + X_{Avgdisplacement}, Y_i + Y_{Avgdisplacement})$. For Each received packet, the server calculates the device's position and compares it to the covered area limits to eventually adjust the estimated position. In fact, our algorithm takes in account the change of the direction of the mobile device and adjusts the position with a distance calculated according to the mobility model. For example in the case of a Zigzag model (Fig. 2), the position is adjusted each time Y_i exceeds Y_{min} or Y_{max} or X_i exceeds Gap (Fig.2).

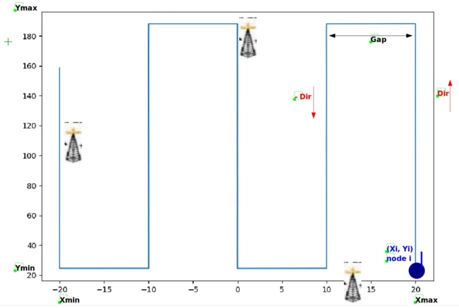


Fig. 2: Zigzag mobility model example

According to the n^{th} estimated position, the server will decide if the device should modify its configuration depending on the calculated change rate that will be defined in the next section.

B. Configuration change rate

We assume that the configuration change should be done at each n packets ($1 < n < 20$). Indeed, the choice of n very small (equal to 1) generates a significant extra cost in the configuration change, while n high (for example 20) causes the loss of packets sent with inappropriate parameters. However, the choice of n depends on the devices mobility. After the estimation of the n new positions of a device, the server calculates a configuration change rate according to equation 4, where $RSSI_{real}(n)$ is the real RSSI measured for the n^{th} received packet, $B_{inf}(n)$ corresponds to the lower bound of the RSSI limit range of the current mode and $RSSI_{estim}(2n)$ is the estimated RSSI of the $2n^{th}$ packet.

$$\mathrm{R}_{change} = \frac{|RSSI_{real}(n) - RSSI_{estim}(2n)|}{|B_{inf}(n) - RSSI_{estim}(2n)|} \tag{4}$$

If the configuration change rate is equal to or greater than a fixed threshold $\alpha(\alpha = 0.5$ in our case), it means that the device is in a new zone and it is necessary to change its configuration mode. Otherwise, the mode does not change.

C. Choice of new configuration mode

To decide to which mode a device configuration should change, we calculate a transition rate $R_{A \to B}$ from mode "a" to each other mode "b" according to equation (5). This rate defines the degree of belonging of the estimated RSSI of the 2^{th} packet in the interval $[B_{inf}, B_{sup}]$ of each configuration mode. B_{inf} defines the lower bound of the old mode (a) and B_{sup} is the upper bound of the target mode (b). Finally, the server will choose the mode with the biggest rate in the raw presented in equation (6). The new configuration (mode) will be communicated to the device through the LinkADRReq command message.

$$\mathrm{R}_{A \to B} = \frac{1/2 \times |B_{sup}(b) - B_{inf}(a)|}{|B_{sup}(b) - RSSI_{estim}(2n)| + |B_{inf}(b) - RSSI_{estim}(2n)|} \tag{5}$$

$$R_{A \to B} = \max(R_{a,1}, R_{a,2}, \cdots, R_{a,9}, R_{a,10}) \tag{6}$$

V. EXPERIMENTATION RESULTS

To evaluate our proposed E-ADR, we have conducted several experiments using the Waspmote-SX1272 devices and gateways [4], [5]. We have integrated a field in the LoRa Join Request frame where the device indicates to the server what mobility model will be used and the area to cover. We have tested several use cases (Cleaning robot, Drones for inspections, Monitoring robot, Feeding system and temperature sensor) in a smart farm environment with fixed and mobile devices.

We assume that we have 3 gateways (A, B, C) whose positions are respectively: G_A[18.02 , 8.62], G_B[-28.31 , 45.14], G_C[-24.25 , 115.04]. We consider the case of 5 devices and evaluate the network performance in terms of ToA and energy consumption using equation (7) [22], where I_{tx} is the transmit current for the transmission time T_{tx}, I_{rx} is the receive current for the RX window duration T_{rx}, I_{sleep} is the sleep current for the time spent in sleep T_{sleep} and U the voltage. We evaluate also the packet loss in the case of using E-ADR and basic ADR respectively.

$$E = (T_{tx} \times I_{tx} + I_{w1w} + T_{rx1w} \times I_{rx1w} + I_{w2w} + I_{rx2w} + T_{sleep} \times I_{sleep}) \times U \tag{7}$$

We assume that the server assigns the devices the mode 1 in the join answer to use for the first packets. Also we choose to adapt the configuration every $n = 3$ packets, which gives good results according to our context. In our use case, most of the time the device needs to change its mode every $n = 3$ positions. If we check the allocation every one received packet, it increases the transmission delay and the energy consumption. If we increase the change frequency for a number higher than $n = 3$, we risk more packet losses. In the following, we present and analyze the experimental results for several uses cases.

A. Cleaning robot use case

The farmer needs to have the animal shelters cleaned and checked using a cleaning robot. we take the case of a robot that traverses an area of dimensions ($[Y_{min}, Y_{max}] = [24.6, 188.1]$,$[X_{min}, X_{max}] = [-20, 20]$) and using the zigzag mobility model in Fig. 2. The robot sends 24 messages of 35 bytes each indicating the animal shelter state.

Fig. 3 shows the comparison between the basic ADR and the E-ADR in terms of time on air and energy consumption.

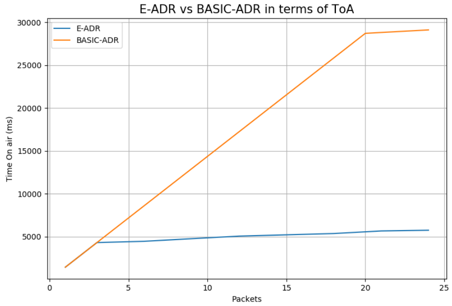
Comparing the basic ADR (red line) to the proposed E-ADR (blue line), we can see the minimization of ToA in Fig. 3a for the cleaning robot. We have minimized 80.26% of the transmission time of the basic ADR. In the same way, the gain on energy consumption is increased about 60.23% (Fig. 3b). This is due to the reduced waiting packets (3 instead of 20) in E-ADR and its quicker configuration re-adaptation capacity.

B. Drone inspecting the parcels

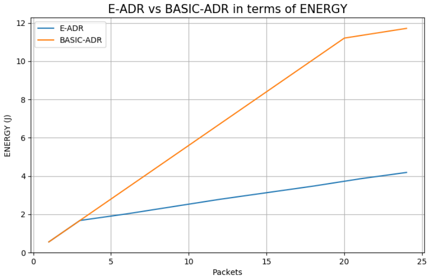
Drones are set up to fly over the parcels using the "Square mobility" model in Fig. 4, covering ($[Y_{min}, Y_{max}] = [24.6, 188.1]$,$[X_{min}, X_{max}] = [-20, 20]$) and sending 55 messages of 143 Bytes containing information about the deteriorated areas, the estimation of damage and the amount of missing product on these parcels. The positioning algorithm will help the farmer to know the positions of the drone in order to detect the damage's zones. Fig. 5 compares the basic ADR to E-ADR in terms of time on air and energy consumption.

Fig. 5 shows that E-ADR decreases the time on air about 81.04% and the energy consumption about 67.82% comparing to the basic ADR.

Thus, it saves the time allowed by the duty cycle limitation for the device and the node will have more chance to send more data. Hence, we remark that E-ADR decreases the packet loss. In fact, our proposal avoids exceeding the duty cycle



(a) Evaluation of Time on Air (ToA)



(b) Evaluation of Energy Consumption

Fig. 3: Case of a mobile Cleaning Robot

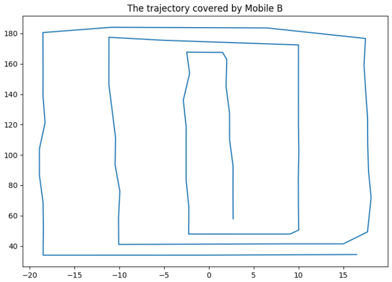
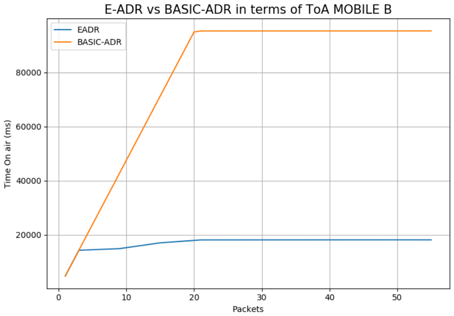


Fig. 4: The trajectory traveled by Device B

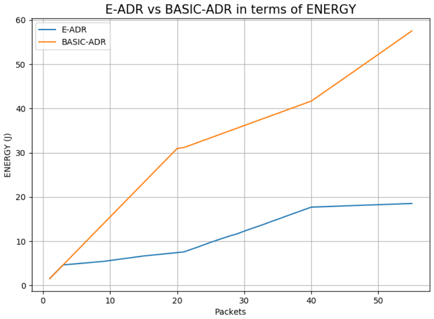
restriction (36s/cycle). In Fig. 5, we see that in the case of basic ADR, the drone can no longer send its data from the 8^{th} packet as it exhausts its allowed duty cycle. This represents a loss ratio of of 87% (we observed only 7 received packets over 55 sent packets). So, E-ADR in this case optimizes the QoS by choosing the right and the optimal configuration.

C. Vegetable and fruit maturity monitoring robot

We take a case of a robot that travels the parcels using the zigzag mobility model presented in Fig. 2, covering ($[Y_{min}, Y_{max}] = [24.6, 188.1]$,$[X_{min}, X_{max}] = [-30, 30]$) and sending 45 messages of 17 Bytes containing information about the maturity of the fruits and vegetables. The farmer saves the coordinates of the areas where the fruits and vegetables are not mature enough, to not send the employees to this zone. Fig. 6 compares the basic ADR to E-ADR in terms of time on air and energy consumption.

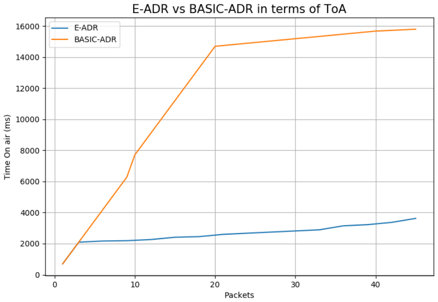


(a) Evaluation Of Time on Air (ToA)

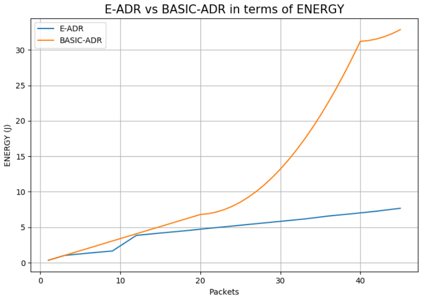


(b) Evaluation of Energy Consumption

Fig. 5: Case of a mobile drone



(a) Evaluation of Time on Air (ToA)



(b) Evaluation of Energy Consumption

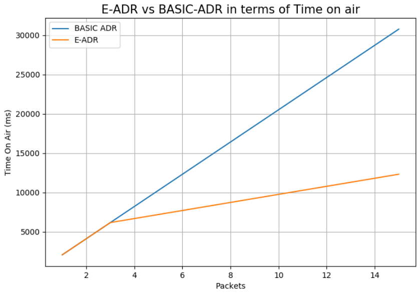
Fig. 6: Case of a robot in a parcel

For this application, the basic ADR allocates a new configuration after each set of 20 received frames when the ADR is set. But this allocation will be done depending on the maximal RSSI received during these 20 frames. We remark that the new parameters allocated after the 20^{th} frame which correspond to mode 7 is not efficient to send the 34^{th}, 35^{th} and the 36^{th} frames (the robot will move away from the 3 gateways and exceeds the limit range of mode 7). The same problem happens after the new allocation done by the server after the 40^{th} frame which corresponds to mode 8. We remark that the 43^{th}, 44^{th} and the 45^{th} frames are not received because the new allocated parameters does not correspond to the robot's needs.

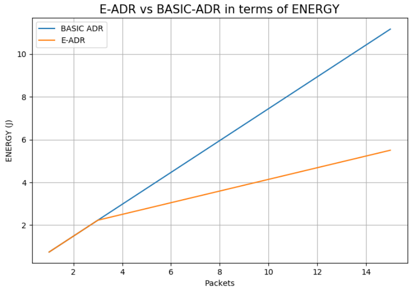
E-ADR however allows the robot to minimize its transmission time about 78.21% and its energy consumption about 45.55% comparing to the basic ADR and helps us to avoid the risk of packet loss that is about 13% in the case of basic ADR (39 packets received over 45 packets sent).

D. *Feeding system and temperature sensor*

An intelligent feeding system is also installed to send 15 messages of 50 Bytes containing the amount of fed animals and their identifiers. We also assume that a temperature sensor is installed for checking the temperature variation in the farm's house by sending 2 messages of 30 Bytes when the temperature becomes higher or lower than a targeting range. Fig. 7 compares the basic ADR to E-ADR in terms of energy consumption and time on air for the feeding system.



(a) Evaluation of Time on Air (ToA)



(b) Evaluation of Energy Consumption

Fig. 7: Case of a feeding system

We remark that E-ADR minimizes the ToA about 60.01% and the energy consumption about 50.7% (See Fig.7).

VI. CONCLUSION

ADR may be used to reconfigure node parameters for adapting to node mobility in LoRa networks, but suffers from low adaptation speed and low performance. E-ADR that we proposed and implemented in Waspmote SX1272 solves those issues. Through experimental tests of different scenarios we have shown that E-ADR outperforms basic ADR in terms of time on air and energy consumption. Moreover, it has reduced or eliminated the packet losses since it minimizes the use of the allowed time limited by the duty cycle. So E-ADR allows LoRa networks to supporting mobility, with known mobility pattern. We are interested in a future work to compare our approach (E-ADR) to other proposals particularly EXP-SF and EXP-AT. In addition, we intend to work on a better accuracy of the device position estimation given its mobility model.

REFERENCES

[1] N.SORNIN (Semtech), A.YEGIN (Actility), ”LoRaWAN Specification v1.1,” October 11, 2017.
[2] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, Analysis of capacity and scalability of the lora low power wide area network technology, 22th European Wireless Conference, May 2016, pp. 16.
[3] LoRa Alliance, LoRaWAN What is it? A technical overview of LoRa and LoRaWAN, November 2015.
[4] Libelium comunicaciones Distribuidas S.L.: Waspmote LoRa 868 MHz 915 MHz SX1272 networking guide, Revision 4.2. (November 2015).
[5] Libelium Comunicaciones Distribuidas S.L.: Waspmote data frame programming guide. version 7.2. (October 2017).
[6] The Things Network, The Thing Network Wiki: Adaptive Data Rate, https://www.thethingsnetwork.org/wiki/LoRaWAN/ADR, 2017.
[7] J. Petaejaejaervi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage, Int. Journal of Distributed Sensor Networks, vol. 13, no. 3, pp. 116, 2017.
[8] D. Bankov, E. Khorov, and A. Lyakhov, Mathematical model of lorawan channel access, in WoWMoM, June 2017, pp. 13.
[9] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, “Poster: A fair adaptive data rate algorithm for lorawan, EWSN 2018.
[10] O. Georgiou and U. Raza, “Low power wide area network analysis: Can lora scale?” CoRR, abs/1610.04793, no., 2016.
[11] M. Bor, U. Roedig, T. Voigt, and J. Alonso, “Do lora low-power wide-area networks scale?,” 19th ACM MSWiM , Malta, 2016, pp. 5967.
[12] M. Bor and U. Roedig, “Lora transmission parameter selection,” 13th DCOSS, Canada, 5-7 June 2017, .
[13] M. Slabicki, G. Premsankar, M. Di Francesco, “Adaptive configuration of lora networks for dense IoT deployments,” NOMS IEEE/IFIP, 23-27 April 2018, ISSN 2374-9709, Taiwan.
[14] F. Cuomo, Ma. Campo, A. Caponi, G. Bianchi, G. Rossini, P. Pisani, “EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations,” 13th IEEE WiMob, 9-11 October 2017, Italy.
[15] D. Kim, S. Kim, H. Hassan, J.H. yukPark, “Adaptive data rate control in low power wide area networks for long range IoT services,” Journal of Computational Science, Vol 22, September 2017, Pages 171-178.
[16] R. Mardeni, Othman, S. Nizam, “Node Positioning in ZigBee Network Using Trilateration Method Based on the Received Signal Strength Indicator (RSSI),” European Journal of Scientific Research, Vol.46, No.1 (2010), pp.048-061.
[17] A.R. Kulaib, R.M. Shubair, M.A. Al-Qutayri, and J.W.P. Ng, “An overview of localization techniques for wireless sensor networks,” IIT 2011, pp. 167172, April 2011.
[18] X. Zhu, Q. Li, and G. Chen, “APT: Accurate outdoor pedestrian tracking with smartphones,” in Proceedings of the 32nd IEEE INFOCOM, pp. 25082516, Italy, April 2013.
[19] “Sigfox Geolocation, http://www.sigfox.com/en/sigfox-geolocation.
[20] H. Tsuji, P. Cherntanomwong, and J.-I. Takada, “Experiential evaluation of outdoor radio source localization using spatial information of array,” APMC 2007, Tailand, December 2007.
[21] B.C. Fargas and M.N. Petersen, “GPS-free geolocation using LoRa in low-power WANs,” GIoTS 2017, Switzerland, June 2017.
[22] Ukko-Pekka Peura, “LoRaWAN optimization for a batery powered sensor network,” Bachelors Thesis, Oulu University of Applied Sciences, 2018.