Optimization of LoRaWAN Access control

Hajer Tounsi¹ · Norhane Benkahla^{1,2} · Ye-Qiong Song^{,2} · Mounir Frikha^{1,3}

Received: date / Accepted: date

Abstract Long Range Wide Area Network (LoRaWAN) enables flexible long-range communication with low power consumption which is suitable for IoT applications. LoRaWAN's performance is due on the one hand to its spreading factor modulation allowing the spread out of communication between end-devices and gateways on different frequency channels and data rates. And on the other hand, to the ability to manage for each node its data rate and its transmission power thanks to the adaptive data rate (ADR) scheme in order to increase the overall network capacity and to maximize the battery life of end devices. However, because of the Aloha access technique adopted by LoRaWAN, the risk of using the same data rate on the same channel is not negligible. Despite the limitation of the duty cycle for each node, the risk of collision is high with the increase of the number of end devices which degrades the LoRaWAN's performance. In this context, our paper proposes different approaches to improve the performance of LoRaWAN. The first contribution consists in improving the ADR technique to meet the characteristics of a mobile environment. The new mechanism proposed, called VHMM-based E-ADR, consists of adapting the data rate of the end-device according to its position. The second contribution consists in better managing the use of the duty cycle by proposing a dynamic sharing mechanism (Dynamic Duty-Cycle).

The last contribution consists in proposing a deterministic access technique to replace Aloha. Our experimental study has shown that our proposals give better results in terms of Packet Delivery Ratio (PDR) and energy consumption than basic LoRaWAN in a mobile environment.

Keywords LoRaWAN, ADR, ALOHA, DC, TDMA, HMM.

1 Introduction

With the rapid growth of the IoT, a promising technology, LoRaWAN (Long Range Wide Area Network Protocol) has emerged in the research field [1]. This technology has the advantages of wide coverage, low cost, low power, simple deployment and support for large number of connections. Based on Long Range (LoRa) physical layer modulation, LoRaWAN enables low throughput communications across long distances with low energy consumption. It operates in the license free ISM bands of 433, 868, and 915 MHz with duty cycle regulation of 1% for each device [1]. To handle different data rates, LoRa defines different configurations according to the customization of several physical layer parameters, such as Spreading Factor (SF), Bandwidth (BW), Transmission Power (TP), and Coding Rate (CR). The network topology of LoRaWAN Network is cellular-like with most of the complexity pushed outside of the capillary network. End devices (EDs) communicate with a central network server through one or more LoRaWAN gateways (GWs). EDs are not associated to a particular GW: the GWs forward all the received messages to the network server which handles messages de-duplication, downlink scheduling and routing of uplink data to the application servers. LoRaWAN is strongly uplink oriented but each uplink transmission is followed by two short receive windows. LoRaWAN offers three service classes: A, B and C depending on

H. Tounsi
✉ hajer.tounsi@supcom.tn

N. Benkahla
✉ norhane.benkahla@supcom.tn

Y-Q. Song
✉ ye-qiong.song@loria.fr

M. Frikha
✉ m.frikha@supcom.tn

¹ University of Carthage, SUP'COM, Mediatron, Tunisia · ² Université de Lorraine, CNRS, LORIA, France · ³ King Faisal University (KFU), Saudi Arabia

the application. The main difference between these classes is how programming their receiving windows. Class A opens two receiving windows at specific time after sending up-link data to the gateway. Therefore, this class consumes minimum energy and must be the default class in all nodes. Class B is used for more down-links, hence it opens more receiving windows which are scheduled periodically by the network server. Whereas, class C nodes have their receiving windows always open except while transmitting, therefore, their energy consumption is the highest among all three classes. To maximize both battery life of the end-devices and overall network capacity, the LoRa network server can manage the data rate and RF output for each end-device individually by means of an adaptive data rate (ADR) scheme. ADR consists in modifying the configuration of the various parameters (SF, BW, TP, etc.) in order to increase the data rate of the end-device and offer it a better QoS. While the LoRaWAN features [1] may ease the deployment of LoRa networks with thousands of nodes by proposing some mitigation mechanisms such as multiple channels, orthogonal spreading factors, dynamic channel discrimination, a LoRa network performance remains limited in some environments especially in case of device mobility, obstacles, high data transmission, etc. This performance degradation is mainly due to the non-adaptability of ADR to the mobility context, to the use of ALOHA as a network access mechanism, and to the duty cycle restriction imposed by the regulation authorities like the European Telecommunications Standards Institute (ETSI) [3]. Indeed, retransmissions of packets lost due to interference (ALOHA) or due to bad rate adaptation (ADR) will quickly cause the Duty Cycle limit (DC) to be exceeded and therefore data to be lost. In this paper, we propose several improvements of these techniques so that they react better to the mobility of nodes. The first contribution deals with the ADR enhancement. Hence we propose a new ADR mechanism called “VHMM-based E-ADR” which adapts the end-device data rate in function of the prediction of its future position while moving [4]. The second approach consists into a new idea to manage the duty cycle [5]. Indeed, some nodes need to send for a period that exceeds the authorized service time while others do not use the totality of the DC. Our contribution is to allow nodes with excess up time to offer it to those who lack it. This technique called dynamic Duty Cycle allows nodes with a large amount of data to be sent per cycle to do so successfully. The last contribution called ” triple joint allocation EADR-Slot-Channel ” consists in replacing Aloha by a deterministic channel allocation technique based on the reservation of the channel, the time slot and the bit rate (SF) with which the transmission will take place. This allocation depends on the optimal configuration performed by VHMM-based E-ADR and the availability of channels and programmed transmission slots. Our proposals were implemented on an experimental

platform made up of LoRa Wasmote SX1272 [6] [7], STM32 [8] and Pycom Lopy4 [9] nodes.

The rest of the paper is organized as follows. Section 2 highlights the VHMM-based E-ADR mechanism and its evaluation compared to the basic ADR in the case of mobility. Section 3 presents the Duty cycle paradigm and how we can improve its use with a dynamic approach. A comparison of the dynamic DC mechanism to the fixed DC is presented. Section 4 describes the triple joint allocation EADR-Slot-Channel contribution and the tested scenarios comparing this new allocation mechanism to the Aloha one. Section 5 resumes the all contributions and concludes the paper.

2 VHMM-based E-ADR mechanism

In order to achieve a high network performance, LoRaWAN uses different data rates provided through the choice of different configuration parameters (SF, BW, TP, CR). The selection of those parameters is a trade-off between communication range, throughput and energy consumption. For example, the increase of SF gives the highest range at the expense of lowest Data Rate (DR), so, higher ToA as well as energy consumption. However, its decrease allows a high DR and low energy consumption at the expense of a reduced range. Table 1 presents the different modes defined by libelium [6]. Each mode specifies a different rate characterized by a given combination of transmission parameters and defines a communication range according to the quality of the received signal.

Table 1 Configuration modes for TP=14dBm [6]

	BW mode (kHz)	CR	SF	RSSI(dB) [Bi , Bs]	DR (Kbps)	SNR (dB)
1	125	4/5	12	[-134, -131]	0,293	-20
2	250	4/5	12	[-131 , -129]	0,585	-19
3	125	4/5	10	[-129 , -128]	0,976	-18.2
4	500	4/5	12	[-128, -126]	1,718	-17.5
5	250	4/5	10	[-126, -125.5]	1,953	-16
6	500	4/5	11	[-125.5, -123]	2,148	-15
7	250	4/5	9	[-123, -120]	3,515	-12.5
8	500	4/5	9	[-120, -117]	7,031	-11
9	500	4/5	8	[-117, -114]	12,50	-10
10	500	4/5	7	[≥ -114]	21,875	-7.5

In order to ensure the QoS requirements of end devices while minimizing energy consumption, LoRaWAN implements an ADR mechanism [10] to adjust the throughput of each node. This mechanism controls the uplink transmission parameters of LoRa nodes (SF, BW, TP, and CR) depending on the link budget in order to

improve the packet delivery ratio (PDR). The main idea of ADR is to manage the DR based on the link budget estimation in the uplink message and the maximum Signal to Noise Ratio (SNR) required for accurately decoding data packets at the existing DR (See Table 1). In fact, for each node, the server re-adapts the configuration mode every 20 received packets by only decreasing the SF (increasing DR) according to the best received SNR [10].

Several experiments reveal that ADR can improve the energy efficiency until a certain extent, but the allocation strategy on which it is based does not allow it to achieve an optimal allocation [11]. In fact, parameters adjustment according to the best SNR only considers perfectly stable conditions. However, environment condition variation (weather conditions, radio interference, obstacles, and mobile nodes) may drastically change the SNR measurement. In addition, waiting the reception of 20 packets before configuration adaptation takes too long time, preventing the quick adaptation to the environmental condition changes in both static and mobile contexts. Moreover, nodes which lost many packets, because they are not in the range of the gateway, can not decrease their data rates only after 64 retransmissions which risks worsening their performance by increasing their transmission time and so the end of their duty cycle. Finally, a major challenge with the ADR mechanism is that LoRa specification does not describe the way the server should instruct nodes regarding rate adaptation, which results in a gap in terms of private ADR implementations. Consequently, many different ADR schemes have been proposed, to improve the network quality of service (QoS).

To deal with ADR limitation in the case of a dynamic environment, we proposed an enhanced ADR mechanism called "VHMM-based E-ADR" [4]. This scheme aims to guarantee the reliability of transmissions (low PLR) face to unstable environments and mobility, while minimizing energy consumption.

2.1 Principle

The objective of our solution is to find the optimal configuration of parameters that will satisfy the device's needs in terms of QoS even in case of random mobility. Thus, we propose to fine-tune SF by both increasing or decreasing it on the server side according to the prediction of the future node position (POI), based on a Variable Hidden Markov Model (VHMM). VHMM-based E-ADR is a proactive scheme while the current ADR is a reactive one, which gives our approach a feature of quick re-adaptivity to condition environment changes. A hidden Markov model (HMM) [12] is a statistical construct that is used to solve prediction problems in many applications. In our case, we used a combination of two HMM (1st order HMM and 2nd order HMM), called VHMM (2,1) to define the future position of a mobile node. We have shown in a previous

work [4] that V-HMM (2,1) is better than HMM-1, HMM-2 or HMM-3 in terms of prediction accuracy.

2.2 Position prediction

2.2.1 HMM principle

The model can be visualized as an interlocking set of states. The hidden Markov model generalizes the observable Markov model because it produces two random variable series; one hidden and the other observable.

- The hidden sequence corresponds to the sequence of states $q_1, q_2, \dots, q_T$, where the q_i take their values among the set of n states of the model $S = s_1, s_2, \dots, s_n$.
- The observable sequence corresponds to the sequence of observations $O_1, O_2, \dots, O_T$, where O_i belongs to $V = v_1, v_2, \dots, v_M$, M observable symbols.

An HMM can be written as $\lambda = (\pi, A, B)$ with:

- A the matrix presenting the transition probabilities between the different states: we define A_{ij} the probability that the model progresses from state i to state j :

$$A_{i,j} = P(q_{(t+1)} = s_j | q_{(t)} = s_i) \quad \forall i, j \in [1 \dots n].$$
- B the matrix presenting the probability of observation of symbols in each state of the model. We define $b_j(k)$ the probability of observing v_k in state s_j : $b_j(k) = P(O_t = v_k | q(t) = s_j)$ where j in $[1 \dots n]$; k in $[1 \dots M]$.

Based on the $A_{i,j}$ and $b_j(k)$ probabilities and using Viterbi algorithm [13], we determine the best path, corresponding to the observation, it means to find in the model λ the best possible future POI. First, we calculate in Eq. 1 the probability of the best path leading to the state i at instant t ($\Gamma_t(i)$). Then, based on ($\Gamma_t(i)$), we deduce in Eq. 2 ($\Gamma_{t+1}(j)$):

$$\Gamma_t(i) = \text{Max}P(s_i, O_t | \lambda) \quad (1)$$

$$\Gamma_{t+1}(j) = [\text{Max}P(\Gamma_t(i), A_{i,j})b_j(O_{t+1})] \quad (2)$$

2.2.2 VHMM-(2,1)

The proposed variable HMM (VHMM-(2,1)) is based on the combination of two HMMs of two orders: HMM-1 and HMM-2. The HMM order defines on how many historical states the next estimated state depends. For example, unlike HMM-1, where only the current state $q(t)$ determines the next $q(t+1)$, HMM-2 depends on the current $q(t)$ and the previous $q(t-1)$. The relevant definitions of VHMM(2,1) are as follows.

Définition 1 *VHMM-(2,1) uses $\lambda = (\pi, A_1, A_2, B_1, B_2)$ to describe the model, where (A_1, B_1) correspond to the previously presented matrix and (A_2, B_2) can be written as:*

- $A_2 = A_{(i-1)i,j} = P(q_{(t+1)} = s_j | q_{(t)} = s_i, q_{(t-1)} = s_{i-1})$ is the probability that there is a 2-order sequence: it means that there is a sequence $q_{(t-1)} \rightarrow q_{(t)} \rightarrow q_{(t+1)}$ in a training sequence.

- $B_2 = b_j(k)$ is the probability that defines the observation probability of symbols in each state of the model. $b_j(k) = P(O_t = v_k | q(t) = s_j)$ where j in $[1...n]$; k in $[1...M]$.

Considering k possible future states defining the possible mobile node directions which are: Direct (D), Back (A), Right (R), Left (L), Left Lean back (AL), Left Lean forward (PL), Right Lean back (AR), Right Lean forward (PR) ($l = \{D, A, R, L, AL, PL, AR, PR\}$).

Eq. 3 presents the probability that at $t+1$, the state $q(t+1)=j$ follows the sequence $q(t-1)=s_{i-1} \rightarrow q(t)=s_i$. If this probability is zero, that's means that there is no sequence corresponding to $[s_{(i-1)} \rightarrow s_{(i)} \rightarrow s_j]$ in the training sets, the server switches to the HMM-1 model and will be based on the probability in Eq 2.

$$\Gamma_{t+1}(j) = [MaxP(\Gamma_t(i), A_{(i-1),i,j})b_j(O_{t+1})] \quad (3)$$

2.3 displacement prediction

Based on the predicted node direction, the server calculates the position coordinates corresponding to this predicted displacement. Equation 4 defines the distance which can be traveled between two different states based on the Speed and the sending period ($P_{sending}$) of the node.

$$d_{distance} = Speed \times P_{sending} \quad (4)$$

The new position of the node is defined from the current position and the predicted direction of its displacement (angle of rotation θ) (Eq. 5).

$$\begin{aligned} X_k &= X_j \pm (d_{distance} \times \cos \theta) \\ Y_k &= Y_j \pm (d_{distance} \times \sin \theta) \end{aligned} \quad (5)$$

2.4 Configuration allocation

After the displacement estimation, the server calculates the distance between the estimated position and the gateways to convert the greatest one to an RSSI value according to Eq 6 [14].

$$RSSI = MeasuredPower - 20\log(Distance) \quad (6)$$

Finally, the server calculates a transition ratio from the actual configuration mode a to each other mode b (b in $\{1, ..., 10\}$). This rate defines the degree of belonging of the estimated RSSI 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 transition rate (Eq. 8). The new configuration (mode) will be communicated to the device through the LinkADRReq command message [1].

$$R_{a \rightarrow b} = \frac{1/2 \times |B_{sup}(b) - B_{inf}(a)|}{|B_{sup}(b) - RSSI_{estim}| + |B_{inf}(b) - RSSI_{estim}|} \quad (7)$$

$$R_{A \rightarrow B} = \max(R_{a,1}, R_{a,2}, \dots, R_{a,9}, R_{a,10}) \quad (8)$$

2.5 Performance evaluation

To evaluate our proposed solution VHMM-based E-ADR, we have conducted several experiments using Waspnote SX1272 [6], STM32 nodes [7] and SX1272 gateways. We consider the 10 predefined modes in Table 1 that will be used in our experimental tests.

Let's consider a LoRaWAN network composed of 5 nodes and 3 gateways. We suppose that nodes are dedicated to animal tracking application in a smart farm [23] and follow random trajectories in an open field S : $S = ([X_{min}, X_{max}] = [-90m, 80m], [Y_{min}, Y_{max}] = [0, 80m])$. Nodes are manually handled and moved at an average speed of $10m/min$.

We suppose that sensors (D_1, D_2, D_3, D_4 , and D_5) are sending respectively: 40 packets, 32 packets, 36 packets, 23 packets, 19 packets at a sending frequency of one packet per minute ($P_{sending}=1min$). Each packet contains 100 Bytes.

2.5.1 Measurements

As mentioned previously, we consider $L = 8$ possible POIs for our VHMM model. The first phase involved the recording of the trajectories directions of nodes over time in order to build a training data base consisting of 115 trajectories for all nodes. These trajectories are divided into groups of 3 POIs for HMM-2 ($q_{(t-1)} \rightarrow q_{(t)} \rightarrow q_{(t+1)}$) and groups of 2 POIs for HMM-1 ($q_{(t)} \rightarrow q_{(t+1)}$).

During the test phase, each trajectory test is repeated 5 times and an average is used to report the different positions of a learned trajectory. We will predict the next POI based on the VHMM-(2,1) model and the trajectories experienced during the training phase.

We define $\alpha_{i,j}$ in Eq 9 as the maximum allowed number of retransmissions of the i^{th} packet of node D_j before the next programmed packet should be sent. In Eq 9, T_{delay} corresponds to the transmission delay of the i^{th} packet and depends on the used configuration mode. $\alpha_{i,j}$ depends also on the node's sent period $P_{sending}$, allowing retransmissions to take place only during $[T_i, T_{i+1})$.

$$\alpha_{i,j} = \frac{P_{sending}(j)}{T_{delay}(i,j)} \quad (9)$$

Moreover, during these $\alpha_{i,j}$ retransmissions and according to the basic ADR mechanism, if the node does not receive an acknowledgment after a threshold $ADRACKLimit$ ($ADKACKLimit = 64$), it will increment its SF to increase the coverage. In addition, during our experiments, if a packet i sent at T_i is still not acknowledged at T_{i+1} , it will be considered as lost. Furthermore, when D_j exceeds the duty cycle limitation of 36sec per hour, the non-transmitted packets will be discarded and considered as lost.

In the next paragraph, we will compare the VHMM-(2,1) based E-ADR to the Basic ADR and 3 ADR variants (ADR+ [15], EXPLoRa-SF and EXPLoRa-AT [16]) in terms of ToA, PLR, and $ADRACKLimit$ variation impact. For ADR+ [15], authors consider the average SNR value instead of the maximal one. In [16], the authors propose two solutions aiming an equitable distribution of the 6 SF values over the registered nodes in the network. The first solution (EXPLoRa-SF) is based on the RSSI values, where nodes having high RSSI values will be allocated a low SF and vice-versa. The second solution (EXPLoRa-AT) aims a fair channel use and is based on the node traffic quantity, where nodes sending a high amount of data will be allocated low SF and vice-versa.

To fairly compare them by focusing only on the algorithm efficiency without the influence of the algorithm execution frequency, we execute the adaptation update after each received packet ($n = 1$) for all the ADR variants.

2.5.2 PLR evaluation

The packet loss rate (PLR) is an important performance criterion since it determines the reliability of the communication, and its increase leads to a large number of retransmissions, which in turn increases the ToA as well as the energy consumption. Moreover, the increase of the ToA leads to a duty cycle exceed, and thus to new losses. The PLR of a node D_j is defined as the number of lost packets (due to bad SF) and the non-transmitted packets (due to exceeding the Duty Cycle) over the total number of packets scheduled to be transmitted by D_j . Note that in the first case, the loss is due mostly to a bad choice of SF which does not guarantee reaching the gateway (no range) or because of a collision with other packets sent at the same time by other nodes using the same channel and

the same SF. Thus, a packet is considered as lost if it is not acknowledged after $\alpha_{i,j}$ retransmissions.

To illustrate the difference in behavior of the different variants, Table 2 shows the mode chosen after the reception of the first packet for each ADR variant.

Table 2 The allocated mode for each node

Node	Basic ADR & ADR+	EXPLoRa-SF	EXPLoRa-AT	VHMM-based E-ADR
D_1	mode 4	mode 2	mode 7	mode 2
D_2	mode 8	mode 8	mode 5	mode 6
D_3	mode 5	mode 3	mode 6	mode 1
D_4	mode 10	mode 8	mode 4	mode 9
D_5	mode 5	mode 2	mode 3	mode 2

Basic ADR and ADR+ are respectively based on the maximum and the average value of the received RSSI list, but since this list will contain only one value ($n = 1$), the server will allocate the same mode each time for these two strategies and it will be, for example *Mode4* for D_1 during the first allocation (Table 2). Concerning EXPLoRa-SF, the 5 nodes will be classified in groups according to their RSSI values, and in our case D_1 is classified in the group corresponding to *Mode2*. Using EXPLoRa-AT, since D_1 sends the largest amount of data, it will be assigned *Mode7*. D_5 , sending the smallest amount of data, will be assigned *Mode3*.

Table 3 presents the average PLRs obtained from the 5 performed tests for each node D_j using the different ADR variants.

We observe that compared to the VHMM-based E-ADR strategy, the different variants result in a large PLR knowing that the update is done after each received packet and retransmissions are enabled. These retransmissions offer a chance for lost packets to be successfully retransmitted, but first in some cases even after $\alpha_{i,j}$ retransmissions, the SF configuration is not adequate to successfully transmit the packet. Also, these retransmissions lead to an increase in ToA causing the Duty Cycle exceed in some cases. When the Duty Cycle is exceeded, the remaining packets will not be transmitted and will be considered as lost, hence a large PLR is obtained using the ADR variants compared to VHMM-based E-ADR for all nodes (Table 3).

Although the configuration update and parameter adaptation are done after each received packet, the fact that Basic ADR, ADR+, EXPLoRa-SF, and EXPLoRa-AT rely on the old RSSI values of the received packets (instead of the future value such as VHMM-based E-ADR) results in a significant PLR.

Indeed, Basic ADR and ADR + are based on the current RSSI so they do not take into account the mobility of the node.

EXPLoRA-SF, based on the allocation of the SF according to the RSSI values of all the registered nodes in the network, without taking into account the RSSI limits (Table 1), is assigned a SF value that, depending on the trajectory of the node, can be under-allocated or over-allocated, which explains the high PLR. Similarly for the EXPLoRA-AT strategy, for which nodes sending the largest amount of data will be assigned a small SF (larger PLR) and nodes sending a small amount of data will be assigned a large SF (larger energy consumption), this SF may be non-adequate (under/over-allocated) depending on the location of the node with respect to the gateways.

The VHMM-based E-ADR model, based on the next position of the mobile node, allows an optimal configuration allocation whatever the traveled trajectory. Thus we note an average decrease of 20% of the loss rate for VHMM-based E-ADR.

To better understand the causes of packet loss (Duty Cycle exceeding or bad SF allocation even after $\alpha_{i,j}$ retransmissions), we have traced the evolution of the PLR over time separately depending on the cause of the loss (Figures 1-3). Blue curves present PLR due to bad SF and orange curves present the PLR corresponding to non-transmitted packets because of duty cycle overflow.

We show that during the first transmissions, losses are mainly due to the bad SF allocation despite the $\alpha_{i,j}$ retransmissions attempts. Moreover, we see losses due to the duty cycle once the retransmissions have considerably increased, generating the exceed of the Duty Cycle. In fact, we notice that losses in the latter case are earlier in the case of ADR and its ExploRa variants than in the case of VHMM-based E-ADR. This is mainly because of the misconfiguration of SF which causes retransmissions to grow. However, using a prediction-based strategy built on the VHMM model to estimate the future configuration allows us to have fewer retransmissions and therefore fewer losses. Besides, the losses noted in the case of VHMM-based E-ADR (bad SF) are due to prediction errors (bad choice of the SF according to the predicted position) or to collision between nodes. yet, this PLR is less than 15% for all nodes. Furthermore, the duty cycle exceed loss is due to the large number off retransmissions.

2.5.3 ToA and energy consumption evaluation

Due to the limitation of the duty cycle per node, the busy time of the channel (ToA) is an important factor in the LoRa network. This parameter is closely related to losses. In fact, the more retransmissions there are, the more this time increases and therefore the risk of reaching the DC limit. In addition, the more the ToA increases, the more energy is consumed.

Figures 4-6 present the cumulative ToA consumed by packets transmitted during the one-hour cycle by nodes D_1 , D_2 and D_3 (using a logarithmic scale) for the different ADR variants.

By assigning a non-adequate SF, the nodes using the different strategies face a large number of retransmissions and thus an increase in ToA as well as significant energy consumption (Table 4).

Figures 1-3 confirm the exceeding of the DC by the nodes which noted a lot of loss.

Since VHMM-based E-ADR chooses more optimal configuration, the total ToA is less than the other ADR variants (case of D2). However, in some cases such as D1 and D3, even with VHMM the node re-transmits some unacknowledged packets several times because of a bad choice of SF or a collision with other packets sent with the same configuration on the same channel. Hence, the overflow of DC.

2.5.4 ADRACKLimit variation impact

In order to reduce the retransmissions and the risk to bypass the duty cycle, we can adjust the limit number of retransmissions *ADRACKLIMIT*. Thus, by reducing this value, the node is allowed to reduce its throughput (increase its SF) more quickly, and therefore maximize the chance that its packets reach the gateway.

Tables 5 and 6 respectively present the impact of the *ADRACKLIMIT* on PLR and energy consumption parameters.

The results obtained in Tables 5 show that a low *ADRACKLIMIT* ($m = 8$) provides a higher chance of reaching the optimal SF quickly, as the node increases its SF every *ADRACKLIMIT* retransmissions, thus avoiding a high number of retransmissions, Duty cycle exceed, high PLR, and high energy consumption (Table 6). However, even with a low *ADRACKLIMIT* (e.g., *ADRACKLIMIT* = 8), the ADR variants still consume more energy compared to VHMM-based E-ADR, since they notice more losses and thus more retransmissions.

2.5.5 Synthesis

Through our tests, we show the effectiveness of the VHMM-based E-ADR aiming to provide a good compromise between energy consumption and transmission reliability, minimizing losses, thus avoiding retransmissions and increasing energy efficiency by reducing the ToA. However, in some cases, the nodes need to send more traffic like D_1 and D_3 and use low rate configuration modes (long channel occupancy) during their transmissions due to the need for a large range. This leads to a duty cycle exceed even with the VHMM-based E-ADR solution. In this context, we proposed in [5] a solution for sharing activity time between nodes sending varying amounts of data. This mechanism called "Dynamic duty cycle" is presented in the next section and evaluated in conjunction with VHMM-based E-ADR.

Table 3 Average PLR of the 5 tests processed by each D_j

Node	Basic ADR & ADR+	EXPLoRA-SF	EXPLoRA-AT	VHMM-based E-ADR
D ₁	40%	30%	95%	27.5%
D ₂	37.5%	43.75%	47.83%	12.5%
D ₃	44.44%	44.44%	50%	19.44%
D ₄	26.08%	21.73%	26.08%	8.69%
D ₅	31.57%	26.31%	31.57%	10.52%

Fig. 1 PLR evolution for D_1

Fig. 2 PLR evolution for D_2

3 Dynamic duty cycle mechanism

LoRa operates in the ISM Radio Frequency band [17], on which restrictions are imposed by the regional authority regulations such as ETSI (European Telecommunications Standards Institute). Although, the ETSI allows either the use of Duty Cycle (a maximum time during which a device can transmit) or the procedure called "Listen Before Talk (LBT) [18]", the current LoRaWAN specification [1]

uses the duty cycle technique to be compliant to the current regulation of the European region. In fact this restriction imposes to each node a duty cycle of 1% per cycle, that means 36 sec per hour. If this duty cycle limitation does not bother traditional low data rate IoT applications, it may exclude important applications where some sensors occasionally need more than 36 seconds per hour to transmit their urgent data following burst

Fig. 3 PLR evolution for D_3

Fig. 4 ToA evaluation for D_1

Fig. 5 ToA evaluation for D_2

Fig. 6 ToA evaluation for D_3

events detection. We have shown in the previous section, a scenario where the duty cycle is exceeded (D_1 , D_2 , D_3).

To introduce more flexibility, we propose a dynamic duty cycle mechanism which is based on the duty cycle

sharing. We propose three classes of nodes (classical, donor and requester).

The main idea is to provide "donor" nodes, that do not use all their maximal allowed activity time, the ability

Table 4 The average energy consumption per received packet (J)

Node	Basic ADR et ADR+	EXPLoRA- SF	EXPLoRA- AT	VHMM- based E-ADR
D_1	0.96	0.89	1.22	0.72
D_2	0.63	0.61	1.03	0.47
D_3	0.92	0.88	1.11	0.69
D_4	0.57	0.52	0.55	0.36
D_5	0.49	0.41	0.53	0.31

to offer their unused activity time to "requester" nodes that need to exceed the duty cycle limitation (1%).

3.1 Dynamic Duty Cycle mechanism on the node side

We propose that each node wishing to participate in the duty cycle sharing mechanism should specify in the registration message or in the sent data packet, its role according to the size of the remaining data to be sent during this cycle and therefore the transmission time (ToA) it needs.

We suppose that the number of packets to send is known, so, each node estimates the ToA needed for their transmissions (ToA_E) based on the worst case ($mode1$) [5], to define if the node can be a donor or a requester. Otherwise, it remains classical that means it doesn't participate to the sharing mechanism.

In case of the device is a "Donor" (estimated transmission time < duty cycle), it will also indicate the remaining time it will lend. Otherwise, if it is a "Requester" (estimated transmission time > duty cycle), it specifies the needed time to borrow before starting transmission (Eq 10). We propose to encode the information of the role of the equipment and the calculated time (for the loan or the donation) in the registration message and especially in the RFU and FOpts fields of the MHDR field [1].

$$(T_{Req/Don})time = LimitToA - ToA_E \quad (10)$$

When the server receives the registration message or the packet data, it saves or updates the node information according to the type of each node in the corresponding table.

After the registration phase, each node begins sending its data using the LoRaWAN protocol. The node continues to send information about its participation in the activity sharing mechanism until it is satisfied. In addition, it can potentially modify its offer or request dynamically to adapt to a change in configuration (E-ADR or retransmission due to loss). It means, the time requested or to be given can be adjusted during the transmission according to the used configuration adjusted thanks to VHMM-based E-ADR. Hence, this allows a node that did not participate in the sharing mechanism during the registration phase, and

realizes that it should send more messages following an event, to request additional time during the transmission of its data or it allows a node that has recovered ToA due to a change in SF to become a donor.

When sending a packet P , the node checks if the ToA_{limit} allows it to transmit its next packet using the current configuration mode ($mode(i)$), otherwise its transmission is interrupted. If its ToA_{limit} is sufficient, it calculates the time to recover (T_{gap}) following the sending of the current packet with a mode (i) > $mode1$ according to Eq 11.

$$T_{gap} = (ToA_p(mode1)) - (ToA_p(mode(i))) \quad (11)$$

If the node is still a requestor, it sends its packet by reducing its T_{Req} ($T_{Req} = T_{Req} - T_{gap}$), and it updates its ToA_{limit} by reducing the ToA consumed when transmitting the packet ($ToA_p(mode(i))$). If it is an old requester (already satisfied), it becomes a donor of $T_{Donor} = T_{Gap}$.

If the node is a donor or was a donor before, it sends its packet by updating its T_{Donor} ($T_{Donor} = T_{Donor} + T_{gap}$) and it updates its ToA_{limit} .

Otherwise, it is a classical node which does not participate in the sharing. It only updates its ToA_{limit} .

Figure 7 resumes the node side mechanism.

3.2 Dynamic Duty Cycle mechanism on the server side

Upon receipt of packets, the server first updates the configuration mode according to the VHMM-based E-ADR strategy. Then, the server should proceed to a treatment of the requests of participation in the Dynamic Duty Cycle according to the indications contained in the received packet. If it is about a donor then the server adjusts the global time ($T_{G_{Don}}$) to be given and saves the node identifier in the donors table. Otherwise, if it is a requestor, then the server checks whether the overall time to be given ($T_{G_{Don}}$) could satisfy its request or not.

Once the request of a given node can be satisfied (the time offered by the Donors is sufficient), the server updates the remaining activity time of the Donors and grants the additional time to the Requestor by sending an update message (ACK) confirming the allocation of the additional activity time. Otherwise, the ACK message specifies that the grant is not satisfied.

3.3 Performance evaluation

To further improve the VHMM-based E-ADR solution in the case of duty cycle exceed due to the transmission of high traffic at low data rate (high SF, long channel occupancy time), we proposed to combine the two solutions (VHMM-based E-ADR and dynamic DC). In this

Table 5 The *ADRACKLIMIT* (m) impact on the PLR

Node	Basic ADR & ADR+		EXPLoRA-SF		EXPLoRA-AT		E-ADR	
	$m = 8$	$m = 64$	$m = 8$	$m = 64$	$m = 8$	$m = 64$	$m = 8$	$m = 64$
D_1	15%	40%	12.5%	30%	57.5%	95%	0%	27.5%
D_2	25%	37.5%	21.87%	43.75%	25%	47.83%	0%	12.5%
D_3	25%	44.44%	22.22%	44.44%	27.77%	50%	0%	19.44%
D_4	16.66%	26.08%	13.04%	21.73%	13.88%	26.08%	0%	8.69%
D_5	21.05%	31.57%	15.78%	26.31%	15.78%	31.57%	0%	10.52%

Table 6 The *ADRACKLIMIT* (m) impact on the average energy consumption per received packet

Node	Basic ADR & ADR+		EXPLoRA-SF		EXPLoRA-AT		E-ADR	
	$m = 8$	$m = 64$	$m = 8$	$m = 64$	$m = 8$	$m = 64$	$m = 8$	$m = 64$
D_1	0.71	0.96	0.63	0.89	0.94	1.22	0.58	0.72
D_2	0.55	0.63	0.53	0.61	0.86	1.03	0.47	0.47
D_3	0.79	0.92	0.61	0.88	0.92	1.11	0.52	0.69
D_4	0.44	0.57	0.39	0.52	0.42	0.55	0.36	0.36
D_5	0.4	0.49	0.36	0.41	0.38	0.53	0.31	0.31

Fig. 7 The update the DC participation status of a node

section, we evaluate the performance of the dynamic duty cycle mechanism with the use of VHMM-based E-ADR in terms of PLR.

We consider the same scenario described in section 2.5.1. Indeed, the evaluation of this scenario in the case of VHMM-based E-ADR has shown that nodes D_1 and D_3 occupy the channel for a long time and exceed the duty cycle resulting in a packet loss increase (caused by this exceed).

The integration of the dynamic duty cycle mechanism allows the node not using its maximum allowed time (36sec) (D_2 , D_4 and D_5) to share its remaining transmission time with the nodes that have to exceed the 1% duty cycle restriction (D_1 and D_3) in order to provide a better overall QoS.

3.3.1 PLR Evaluation

We present in this section PLR caused by the duty cycle exceed for the different nodes D_1 , D_2 , D_3 , D_4 , and D_5 in the two cases: "VHMM-based E-ADR with basic DC" and "VHMM-based E-ADR with Dynamic DC".

Table 7 presents the PLRs obtained in both cases. We remark that thanks to the dynamic duty cycle, D_1 and D_3 nodes no longer note a loss due to exceeding the duty cycle. Furthermore, the PLR due to duty cycle exceed is decreased in the case of basic ADR and the other variants. Figures 9 and 10 illustrate respectively the variation of PLR in time caused by the duty cycle exceeding for D_1 and D_3 for all compared ADR variants.

We have shown in this section that the dynamic duty cycle approach joined to the VHMM-based E-ADR solution contributes in the PLR attenuation. However, even if the SF allocation is adequate to the needs of the nodes, the management of the distribution of these SFs over the LoRaWAN channels is not considered. Using the same SF on the same channel would cause a significant

Fig. 8 Duty cycle management at the server side

Table 7 PLR caused by the duty cycle exceed

	D_1	D_2	D_3	D_4	D_5
VHMM-based E-ADR	15%	0%	8.33%	0%	0%
VHMM-based E-ADR & Dynamic DC	0%	0%	0%	0%	0%

Fig. 9 Evolution of D_1 PLR due to DC exceeding

increase in the probability of collisions in the case of a dense LoRaWAN network (multiple LoRa nodes). Indeed, all the variants of ADR among others VHMM-based E-ADR modify the configuration of a node independently of the other nodes, that's why the risk of having the same parameters (channel and SF) for several nodes increases as well as the risk of collision and therefore loss. So, in this context, a new channel access mechanism is proposed. This one is based on the channel and slot allocation according to VHMM-based E-ADR choice of SF.

4 EADR-Slot-Channel approach

LoRaWAN was originally designed to be used as a large-scale star network based on Aloha access. Thus, each node transmits its data as soon as it is ready independently of the others and by randomly selecting a channel [22]. This leads to a high probability that simultaneous transmissions use the same channel especially in a dense network. In fact, simultaneous transmissions using the same channel and the same configuration (same SF) will result in the collision of packets unless one of these transmissions uses a much higher TP according to the SNR condition (the signal is received at least 6 dB above the other signals) [20]. This will eliminate all other transmissions and will be the

Fig. 10 Evolution of D_3 PLR due to DC exceeding

only one to be received. This phenomenon is called the capture effect and has been studied in [21].

In addition, LoRa nodes use the Class A service to send their data. In this case, the node transmits firstly UL (Up Link) data to the gateway and then opens two receive windows (RX1 and RX2), giving the gateway the ability to send a DL (Down Link) transmission. This increases the risk of collision with uplink transmissions from other nodes. Indeed, the LoRa nodes do not listen to the channel before transmitting.

Despite the duty cycle restriction, this collision problem constitutes a serious bottleneck for the deployment of LoRa in dense networks, such as in agriculture (connected farm), where hundreds of nodes are envisioned to monitor weather conditions and remotely controlling irrigation and lighting systems.

In this context, we propose a more efficient solution than P-ALOHA to avoid collisions through the scheduling of packet's transmission by the server based on the choice of the slot and the transmission channel according to the chosen configuration. Hence, as TDMA mechanism, our proposal consists in reserving the slot in the first available channel that will be used to transmit the next packet according to its configuration parameters performed thanks to VHMM-based E-ADR.

The information about the new deterministic allocation (EADR-Slot-Channel) will be sent to the concerned node in the ACK_{config} packets (used previously for rate adaptation). The goal of this dynamic approach is to eliminate collisions while preserving energy and minimizing latency.

4.1 Slot and channel reservation mechanism

In accordance with the VHMM-based E-ADR proposal, each time a packet arrives from a node i , the server will define the new configuration (SF, data rate) which will be used for the transmission of the next packet. But instead of the node being responsible for sending this last packet on one of the randomly chosen channels, we propose that the server defines the channel on which the packet must be sent and the corresponding channel slot.

Figure 11 gives the slot/channel allocation algorithm executed by the server. The server has a table of N_{Ch} LoRa channels to be used for each configuration mode ($mode$). Each table ($T_{Chmode\mu}$) contains information about the slots (start and end of the slot) assigned to the nodes using the $mode\mu$ and on which channel. These tables will be used to allocate a slot in an available channel for the next packet.

From the identifiers ($DevEUI$ and $AppEUI$) of each node received in the registration packet (Join Request), the server determines the number of nodes registered in the network (n_D), the total number of packets scheduled for transmission (N_{Tj}) as well as the sending frequency ($P_{sendingj}$) of each node (j).

It should be remembered that all the information necessary for estimating the next slot for a given node is already sent during the join procedure or during the update of the configuration like in the previous proposed mechanisms.

On receipt of each packet from node j , the server estimates the configuration mode ($mode(j)$) for sending its next packet ($i+1$) and defines the next slot to program for its transmission according to Eq 12 and Eq 13 where $P_{sendingj}$ is the sending period of node j , $ToA(i,j)$ mode

is the transmission time of packet i using mode μ and t_{RX} is the duration allocated for the DL transmission.

$$t_{(i,j)start} = t_{(i-1,j)end} + P_{sending(j)} \quad (12)$$

$$t_{(i,j)end} = t_{(i,j)start} + ToA_{(i,j)mode\mu} + t_{RX} \quad (13)$$

Based on the estimated mode ($mode\mu(j)$) and the calculated slots ($[t_{(i,j)start}, t_{(i,j)end}]$) for the next n packets, the server looks in the $T_{Chmode\mu}$ table for the first available channel which can program this slot (Algorithm 1). If the server finds an available channel $C_{lmode\mu}$ (with $l = 1 \dots N_{Ch}$), it assigns it to the slot and updates the Table $T_{Chmode\mu}$ by setting the selected channel $C_{lmode\mu}$ to "unavailable" until $t_{(i,j)end}$. Otherwise, the server looks for the first free slot among the different available channels and chooses it to delay the transmission of the next packet (Algorithm 2). We assume that the shift interval does not influence the configuration allocation.

This information will be sent in the acknowledgment using a new MAC command with CID=0x1B called *NewChannelSlot* in the *FOpts* field [1].

By receiving updates from the server, the nodes update their configuration modes, slot information, and channels to use for their next n packets transmission.

Algorithm 1 : Search for available channel for transmission during τ using $mode\mu$ for a node j

input : $\tau = [t_{(i,j)start}, t_{(i,j)end}]$: a calculated slot for packet i of node j ; $mode\mu$: a selected mode; $T_{Chmode\mu}$: slots and channels table for mode μ ; N_{ch} : number of channel per mode

output : C_l (Available channel during the selected slot)

```

1 for  $v = \tau$  do
2   for  $w \leftarrow 1$  to  $N_{Ch}$  do
3      $X = T_{Chmode\mu}[v, w]$ 
4     if  $X == 0$  then
5        $C_l = w$ ; Update  $T_{Chmode\mu}[v, w]$ 
6         ( $T_{Chmode\mu}[v, w] = 1$ ); exit
7     else if  $X \neq 0$  then
8        $w = w + 1$ 
9   end
10 if  $C_l \neq 0$  then
11   Allocate  $C_l$  to node  $j$  during a slot  $\tau$ 
12 else if  $C_l == 0$  then
13   Search for the first available channel for the first
    next available slot (See Algorithm 2)
```

4.2 Extra (burst) packets detection

It should be noted that each node j can only send its packet at the instant specified by the server (allocated slot)

Algorithm 2 : Search for the first next available channel using $mode\mu$ within its slot τ for a node j

input : $\tau = [t_{(i,j)start}, t_{(i,j)end}]$: a calculated slot for packet i of node j ; $mode\mu$: a selected mode; $T_{Chmode\mu}$: slots and channels table for mode μ ; N_{ch} : number of channels per mode; $N_{\tau\mu}$: number of slot in a channel using $mode\mu$

output : C_l (Available channel during a selected slot)

```

1 for  $v \leftarrow (\tau + 1)$  to  $N_{\tau\mu}$  do
2   for  $w \leftarrow 1$  to  $N_{Ch}$  do
3      $X = T_{Chmode\mu}[v, w]$ 
4     if  $X == 0$  then
5        $C_l = w$  &  $\tau = v$ ; Update  $T_{Chmode\mu}[v, w]$ 
6         ( $T_{Chmode\mu}[v, w] = 1$ ); exit
7     else if  $X \neq 0$  then
8        $w = w + 1$ 
9   end
10 if  $C_l == 0$  then
11    $v = v + 1$ 
12 else if  $C_l \neq 0$  then
13   Allocate slot  $\tau$  using  $C_l$  to node  $j$  for its  $i^{th}$ 
    transmission; exit
```

according to the sending frequency ($P_{sending(j)}$) specified during the joining phase or known by the server according to its identification. However, in case of irregular traffic which causes a modification in the expected sending rate, the node must inform the server when sending its last packet about the change in the time of sending of the next packet. This prevents the server from programming late slots which will cause delays and therefore optimize the node QoS.

4.3 Performance evaluation

This section presents the experimental results of the EADR-Slot-Channel dynamic allocation scheme. We evaluate the performance of our contribution and its efficiency in the context of a static and mobile nodes in a LoRaWAN network. To validate our contribution, the performance is evaluated in terms of PLR and energy consumption.

4.3.1 Measurements

In order to densify the network and increase the risk of interference, we add 10 fixed Lopy [9] nodes to the previous scenario 2.5.1. We limit our choices in this scenario to only 3 channels to create a situation where several nodes would be forced to use the same channel with the same SF which leads to collision phenomenon.

The various nodes transmit their packets as specified in Table 8.

Fig. 11 Slot/Channel allocation by a server

We assume the different nodes register with the server through sending the Join Request message on different time slots using *mode1* in order to guarantee the receipt of the message and avoid join message collisions. When receiving the join message, and based on the ID of the application to which the node is dedicated, the server takes note of the type of node, number of packets to be transmitted, type of application, frequency of sending, etc.). In the case of fixed nodes, the RSSI of the received Join Request will define the configuration mode used (which will be checked after each $n = 1$ received packets to prevent obstacles presence). However, the configuration modes of mobile nodes are variable according to their movements and will be assigned according to the VHMM-based E-ADR strategy after each received packet ($n = 1$).

4.3.2 Packet loss rate evaluation

We have noted in the previous experiments that thanks to VHMM-based E-ADR (Table 3, Figures 1-3) and Dynamic DC mechanisms (Table 7, Figures 9-10), the PLR of nodes D_1 to D_5 have decreased considerably (see eliminated for losses due to DC limitation). Thus, any loss that will be noted in the new scenario is mainly due to collisions caused by the simultaneous sending of several nodes on the same channel and with the same configuration (same SF).

Nodes	type	Packets amount		$P_{sending(i)}$
		regulars	irregular	
D_1	mobile	40	0	1pkt/1min
D_2	mobile	32	7	1pkt/1min
D_3	mobile	36	4	1pkt/1min
D_4	mobile	23	4	1pkt/1min
D_5	mobile	19	5	1pkt/1min
D_6	fixed	15	0	1pkt/1min
D_7	fixed	12	3	1pkt/1min
D_8	fixed	8	4	1pkt/3min
D_9	fixed	11	0	1pkt/2min
D_{10}	fixed	10	0	1pkt/3min
D_{11}	fixed	9	3	1pkt/3min
D_{12}, D_{13}	fixed	13	0	1pkt/3min
D_{14}	fixed	7	0	1pkt/6min
D_{15}	fixed	5	0	1pkt/3min

Table 8 Nodes characteristics

Moreover, because we use the configurations of Table 1, where the power $TP = 14dB$, the capture effect [21] is not applicable. When nodes collide (interfere), packets are not received and must be retransmitted.

Among the various tests carried out (10 tests), we have presented in Figures 12-13 one of these tests scenario in order to reproduce the behavior of respectively the ALOHA approach conducted with VHMM-based E-ADR (EADR-ALOHA) and the deterministic approach

combining VHMM-based E-ADR with the channel and slot allocation (EADR-Slot-Channel).

Figure 12 presents the EADR-ALOHA scenario. We recall that in this case, the channel number to be used is selected in a pseudo-random way between 1 and 3 (3 available channels) and the packets are sent directly following their detection (without any channel listening). On the figure we represent the channel on which the packet is sent by a number (1 to 3) with a color illustrating the mode according to the presented legend.

We remark that many packets are sent by different nodes using the same configuration *mode* (same color in Figure 12), and simultaneously on the same channel (numbered from 1 to 3). These packets interfere and are not received since nodes use the same power $TP = 14dBm$ (No capture effect) [21]. This increases the packet loss rate (PLR).

For example, we consider the packets sent at t_7 by D_1 , D_3 , D_6 , D_{12} , using *mode6* (Figure 12). The pseudo random selection leads the use of *Channel2* by the 4 nodes, causing interference and inducing their retransmissions. Figure 13 presents the results of the EADR-Slot-Channel allocation. We see that the EADR-Slot-Channel approach has allowed at time t_7 , for example, to transmit 3 packets on different channels and to schedule the 4th transmission of D_{12} on another slot on the 1st next available channel, which is *number3*, thus eliminating the risk of collision.

Figures 14-18 show the average PLR of the 10 tests as a function of the number of packets in 3 cases: the scenario of 5 nodes implementing VHMM-based E-ADR and dynamic DC with ALOHA, the scenario of 15 nodes implementing these same mechanisms and the scenario of 15 nodes implementing the deterministic approach (EADR-Slot-Channel). We notice from these curves that by increasing the load of the network (addition of nodes D_6 to D_{15}), the PLR increases because of the collisions between the nodes. However, the scenario using the EADR-Slot-Channel approach notes the lowest loss rate thanks to the elimination of node collisions by programming different send slots for the same configuration. But, this PLR is not zero. This is due to prediction errors which lead to a bad choice of SF.

4.3.3 Evaluation of channel utilization

Load balancing in channels is an important criterion. It is designed to ensure high availability, reliability and scalability of the network. It can improve the packet reception rate and increase network performance by fully utilizing the available resources. The goal is to balance the traffic on different channels to reduce collisions and extend the lifetime and availability of channels without facing gateway DC restriction. To evaluate this criterion, we present in Figure 19 the channel occupancy rate obtained from Figures 12-13.

We observe in this figure that the channel utilization rate is variable with a larger margin difference in the case of "EADR-ALOHA" scenario compared to "EADR-Slot-Channel" due to the pseudo-random channel selection. This unbalanced distribution increases the probability of collisions in highly loaded channels inducing an increase in the PLR. The EADR-Slot-Channel combination achieves a certain level of fairness by balancing the traffic equally across all the available channels and results in an average occupancy rate of 33% with a variability of $\approx \pm 2\%$ in this example.

4.3.4 ToA and energy consumption evaluation

The transmission time (ToA) as well as the energy consumption are important criteria in the optimization of the LoRaWAN protocol QoS. Indeed, the idea is to reduce the $T_{collision}$, but while ensuring an optimal energy consumption. Colliding incoming packets are all retransmitted until they are correctly received in order to reduce the PLR. These retransmissions result in increased ToA and consequently high energy consumption.

In this section, we evaluate the ToA consumed by all nodes when using the different mechanisms in order to validate the efficiency of the "EADR-Slot-Channel" approach. Tables 9 and 10 respectively present the average total ToA per node and its per packet energy consumption in both cases.

Table 9 The average ToA consumed by a node

	EADR-ALOHA	EADR-Slot-Channel
ToA (ms)	34657.08	29732.11

Table 10 Average energy consumption per received packet (J)

	EADR-ALOHA	EADR-Slot-Channel
$Energy_{average}$ (J)	0.197	0.082

When facing a significant PLR due to collisions, nodes using EADR-ALOHA experiences the retransmission of all the interfered packets. Although these retransmissions offer a chance for packets to be correctly received, they significantly increase the ToA (34657.08ms) and consequently the average energy consumption per packet (0.197J). On the other hand, we observe that the ToA consumed using our EADR-Slot-Channel approach is decreased with the mitigation of the collision probability. Indeed, the deterministic slot assignment allows to discard retransmissions and thus to benefit from a low ToA consumption (29732.11) and consequently a better energy efficiency which offers a gain of more than 50%.

Fig. 12 EADR-ALOHA

Fig. 13 EADR-Slot-Channel

4.3.5 Transmission delay evaluation

The transmission delay can be impacted by different factors, for example, the delay of the programming slot. Table 11 presents the average transmission delay per packet for the two tested scenarios.

Table 11 Average transmission delay per packet (TD) (ms)

	EADR-ALOHA	EADR-Slot-Channel
TD (ms)	2173.79	1071.29

Although the EADR-Slot-Channel approach shifts transmission slots when no channel is available, but the average transmission delay is still less than that of EADR-ALOHA. In fact, the slot delay is negligible compared to the delay caused by retransmissions in the case of EADR-ALOHA. In summary, in addition to the significant collision rate obtained by using EADR-ALOHA, the average packet transmission delay is also increased because of the retransmissions of the interfered packets, whereas our EADR-slot-Channel approach allows to reduce the transmission delay by avoiding collisions (thus retransmissions).

Fig. 14 Collision rate caused by ALOHA for D_1

Fig. 15 Collision rate caused by ALOHA for D_2

Fig. 16 Collision rate caused by ALOHA D_3

Fig. 17 Collision rate caused by ALOHA for D_4

Fig. 18 Collision rate caused by ALOHA for D_5

5 Conclusion

The increasing deployment of IoT environments and the use of IoT applications in various domains faces the main challenge of "quality of service (QoS) improvement" in terms of reliability and energy efficiency. So, the objective of this paper is to propose solutions to meet the need in terms of QoS on one side and optimize the resources as

well as the energy consumption on the other side. In this context, our first contribution focused on the definition of an improvement of the rate adaptation mechanism in a context of unknown mobility. The proposed mechanism, VHMM-based E-ADR, provides a solution for computing a new configuration to mobile nodes based on their future displacements defined through a variable hidden markov model (VHMM-(2,1)) and a supervision of the

Fig. 19 Average channels occupancy rate evaluation

corresponding RSSI values with respect to the best rate to be offered. Simulation results performed for different scenarios have shown that our solution is better than other ADR variants proposed in the literature in terms of PLR and energy consumption. Indeed, VHMM-based E-ADR significantly reduces the losses due to a bad choice of SF while improving the energy efficiency.

In a second part, we found that often, for high amount of data transmissions, the losses are due to exceeding the Duty Cycle restriction. Thus, our second contribution is the proposal of a dynamic Duty Cycle sharing mechanism, dealing with the borrowing of the unconsumed and unusable Duty Cycle by a given node in order to allow another to exceed its Duty Cycle. This mechanism allowed to increase the number of packets sent by each node of the network, thus improving the global QoS. Its implementation with the VHMM-based E-ADR improves the QoS performance by further reducing losses due to duty cycle exceed.

Although the proposed E-ADR solution provides better reliability due to the optimal distribution strategy of these parameters over all the nodes of the LoRa network, a dense deployment can lead to collisions due to the use of a same random channel by many nodes and with the same configuration according to only the node position. Our third contribution was the proposal of a deterministic access mechanism based on dynamic slot scheduling according to the dynamic TDMA principle with Round Robin channel choice. This contribution has been named "EADR-Slot-Channel approach" allows to avoid collisions, thus avoiding packet retransmissions and offering a better energy efficiency. All our contributions have been implemented on a platform composed of LoRa nodes (Lopy, STM, etc.) and three Gateways. Despite the improvements made by our proposals reducing the PLR while guaranteeing lower energy consumption and acceptable delays, it should be noted that more precision must be proposed for the prediction of the position of mobile nodes directly affecting the choice of the optimal configuration. We therefore propose to study other prediction methods to have better precision. Otherwise, our study based on real experiments is limited by the number of available nodes. So to study the scalability

of our solutions, we plan to implement our work on a simulator such as LoRaSim [26].

Conflicts of interest/Competing interests

The authors declare there is no conflicts of interest and competing interests

References

1. L. Alliance. (2016). LoRaWAN specification, v1.0.2.
2. R. Nelson and L. Kleinrock. (1984). The spatial capacity of a slotted ALOHA multi-hop packet radio network with capture. In: IEEE Trans. Comm., vol. 32.
3. ETSI. (May 2012). ETSI EN 300 220-1 V2.4.1. Available: www.etsi.org/deliver/etsi-en/300200-300299/30022001/02.04.01-60/en-30022001v020401p.pdf.
4. N. Benkahla, H. Tounsi, YQ. Song, M. Frikha. (2021). VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns. In: the 16th IWCNC conference. China.
5. N. Benkahla, H. Tounsi, S. Ye-Qiong, M. Frikha. (2018). Enhanced dynamic duty cycle in LoRaWAN network. In: Adhoc Now conference. France.
6. Libelium comunicaciones Distribuidas S.L. (2015). Waspnote LoRa 868 MHz 915 MHz SX1272 networking guide.
7. Libelium Comunicaciones Distribuidas S.L. (2017). Waspnote data frame programming guide.
8. Semtech Corporation, STMicroelectronics. (2019). Discovery kit for LoRaWAN, Sigfox, and LPWAN protocols with STM32L0 B-L072Z-LRWAN1.
9. LoPy. (2020). Available: <https://docs.pycom.io/gitbook/assets/specsheets/Pycom-002-Specsheets-LoPy4-v2.pdf>.
10. N. The Things Network. (2020). LoRaWAN Adaptive Data Rate.
11. N. Benkahla, H. Tounsi, YQ. Song, M. Frikha. (2020). Review and evaluation of ADR enhancements for LoRaWAN networks. Telecommunication systems Journal.
12. P. Hofmann, Z. Tashman. (2020). Hidden Markov Models and Their Application for Predicting Failure Events. In: Computational Science – ICCS 2020. Lecture Notes in Computer Science, vol 12139. Springer, Cham. <https://doi.org/10.1007/978-3-030-50420-5-35>
13. D. Suviha, M. Vijayalakshmi, and P. Sameer. (2018). Traffic Prediction Using Viterbi Algorithm in Machine Learning Approach. DOI: 10.1007/978-981-10-8657-1-25.
14. E. Goldoni, A. Savioli, M. Risi, and P. Gamba. (2010). Experimental analysis of RSSI-based indoor localization with IEEE 802.15.4. In: European wireless proceeding. Italy. pp 71–77.
15. M. Slabicki, G. Premsankar, and M. Di Francesco. (2018). Adaptive configuration of loRa networks for dense IoT deployments. In: Network Operations and Management Symposium. Taiwan.
16. F. Cuomo, Ma. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani. (2017). EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In: the 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Italy.
17. LoRa Alliance Technical committee. (2016). LoRaWAN - Regional Parameters, 1.0, LoRa Alliance, Inc., Jul. 45 pp.
18. J. Ortín, M. Cesana and A. Redondi. (2019). "Augmenting LoRaWAN Performance With Listen Before Talk," in IEEE Transactions on Wireless Communications, vol. 18, no. 6, pp. 3113-3128. doi: 10.1109/TWC.2019.2910512.
19. Beltramelli, Luca et al. (2021). LoRa Beyond ALOHA: An Investigation of Alternative Random Access Protocols." IEEE Transactions on Industrial Informatics 17. pp. 3544-3554.

20. C. Goursaud, and J-M. Gorce. (2015). Dedicated networks for IoT: PHY/MAC state of the art and challenges. In: the endorsed transactions on Internet of Things (EAI). Vol 1, pp 1-11.
21. D. Croce, M. Gucciardo, I. Tinnirello, D. Garlisi, and S. Mangione. (2017). Impact of spreading factor imperfect orthogonality in loRa communications. In: the International Tyrrhenian Workshop on Digital Communication. Switzerland. pp 165-179.
22. D. Bankov, E. Khorov and A. Lyakhov. (2016). On the Limits of LoRaWAN Channel Access. In: the International Conference on Engineering and Telecommunication (EnT). pp. 10-14, doi: 10.1109/EnT.2016.011.
23. Semtech Corportation, Smart agriculture. Available: <https://www.semtech.com/lora/lora-applications/smart-agriculture>.
24. D. Falconer, F. Adachi and B. Gudmundson. (1995). Time division multiple access methods for wireless personal communications. In: IEEE Communications Magazine, vol. 33, no. 1, pp. 50-57. doi: 10.1109/35.339881.
25. LoRa Alliance. (2018). LoRaWAN Application Layer Clock Synchronization Specification v1.0.0.
26. T. Voigt and M. Bor. LoRaSIM : <https://github.com/adwaitnd/lorasim>.