VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns

Norhane Benkahla, Hajer Tounsi, Mounir Frikha, Ye-Qiong Song

HAL Id: hal-03283215
https://hal.science/hal-03283215

Submitted on 9 Jul 2021

VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns

Norhane BENKAHLA
MEDIATRON SUP'COM - LORIA
Carthage university - University of lorraine
Tunis, Tunisia - Nancy, France
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—Long Range Wide Area Network (LoRaWAN) introduces the Adaptive Data rate (ADR) mechanism [1] aiming to maximize both battery life of the end-devices and overall network capacity. ADR performs adaptive tuning of radio configurations of the nodes by adjusting bandwidth, spreading factor, coding rate and transmission power parameters whenever the signal quality changes. The ADR algorithm was established for stable radio channel environments and is not efficient when conditions dramatically change (e.g. mobility). So, we have previously proposed an Enhanced-ADR (E-ADR) [2] that deals mobile nodes in case of predefined mobility patterns. However, several Internet Of Thing (IoT) applications, such as smart cattle ranching in smart farms [3], require sensors travelling with unknown or undefined trajectories. So, this paper extends E-ADR to unknown mobility pattern. This E-ADR extension, called VHMM-based E-ADR, is based on a Variable order Hidden Markov Model (VHMM) to predict the node trajectory. It has been implemented on Waspmote SX1272 hardware platform. Experimental results show its high efficiency in terms of the packet loss rate (PLR) and the energy consumption.

Index Terms—IoT, LoRaWAN, ADR, Spreading factor, VHMM(2,1), mobility

I. INTRODUCTION

Among the most popular communication technologies used for Internet of Things (IoT), Long Range (LoRa) technology [4] and its Long Range Wide Area Network (LoRaWAN) MAC protocol [1] are widely used for long-range and low power sensor data transfer. IoT is applied in various domains where both static and mobile nodes are used, such as smart farms, habitat monitoring, logistics, and product life-cycle management.The main challenge for LoRaWAN is to ensure a good data transmission QoS (Quality of Service) while minimising nodes energy consumption. For this purpose, the adaptive data rate mechanism (ADR) is used to efficiently set the transmission parameters of each node according to the network state.

A LoRaWAN node can run with different data rates, resulting in different TOA (Time On Air), different power consumption, so, different use of duty cycle. This can be achieved by combining four parameters: Spreading factor (SF), Bandwidth (BW), Coding rate (CR), and Transmission power (TP). There are 6720 possible configurations by combining the different values of these parameters [5]. However, [6] was limited to only 8 data rates (DR0 - DR7) among these

TABLE I: Configuration modes for TP=14dBm [7]

mode	BW (kHz)	CR	SF	RSSI(dB) [Bi, Bs]	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	[≥ -114]	21,875

configurations. In our work we adopted the 10 modes defined in [7] and presented in Table I.

Table I shows that for a given TP, each mode defines the range of RSSI (Received Signal Strength Indicator) values [B_i, B_s] ensuring the correct reception of the packets. An RSSI lower than B_i may result in packet loss, whilst an RSSI higher than B_s induces useless high power consumption.

How to dynamically choose the best data rate is not trivial. The ADR mechanism of LoRaWAN has been designed for a stable network with fixed nodes. Its convergence speed is relatively low and does not cope well with high dynamic networks [1]. Several studies and experiments revealed its limitation in mobile node scenarios [8]–[10] without offering solutions for these cases. So, to take into account mobile nodes, we have previously proposed in [2] an Enhanced ADR (E-ADR). E-ADR aims at self-adapting the node mobility through quickly re-configuration of the parameters face to dynamic network condition changes. It determines the best configuration corresponding to the RSSI value of the sending node, estimated as a function of the its new position. The position calculation is based on the linear regression prediction method. Our experimental tests have confirmed that E-ADR outperforms Basic ADR thanks to its RSSI-based proactive transmission parameter adjustment, that can also use Right RSSI to prevent packet loss. However, E-ADR being designed for known mobility patterns is not suitable for applications where some nodes follow random/unknown trajectories (e.g., cattle ranching). To cope with applications with unknown trajectories, we propose in this paper to extend E-ADR to

VHMM-based E-ADR which consists to predict the next position of the node using Variable order Hidden Markov Models (VHMM).

The main new contribution of this paper is:

- The exploration of HMM (Hiden Markov Model) to predict the next position of a mobile node and the validation of the optimal order (HMM-2) according to several experimentation.
- The proposition of a variable VHMM-(2,1) model combining both HMM-1 and HMM-2 models
- The integration of VHMM-(2,1) to the E-ADR to adjust the LoRaWAN node configuration according to its new position.

The rest of the paper is organised as follows. Section II describes the background of the work. VHMM-based E-ADR is detailed in Section III. Implementations and results are discussed in Section IV by showing how the proposal is adapted to mobile nodes travelling an unknown trajectory. Finally, we conclude in Section V.

II. BACKGROUND

The use of ADR (also called hereafter Basic ADR) is optional but highly recommended by the LoRaWAN specification [1]. ADR runs in two parts using two independent algorithms on the node-side and on the server-side. The node-side reacts to the packet loss (no receipt of acknowledgements) by increasing the SF to increase its radio coverage range after reaching a retransmission threshold (m). The server-side aims to increase the DR (therefore decrease the SF) in order to save energy based on the best RSSI of the last 20 received packets. The main objective is to converge to an "optimal" SF that can ensure the packet reception with a high data rate on the one hand, and minimise the energy consumption on the other hand.

Analysis and implementations of the ADR have previously been undertaken. Studies in [8]–[10] have shown that the current ADR algorithm (Basic ADR) [1] is effective in stable channel conditions where nodes are static but is not suitable in dynamic networks, especially in the case of mobile nodes or the presence of transient obstacles between a node and its gateway. In addition, it has been identified that the performance of Basic ADR is limited in scale because of the duty cycle restriction (1% or not exceeding 36s TOA every cycle of one hour). Also, in a lossy channel, the node-side algorithm convergence time to an optimal configuration mode (SF) is very slow (only after $ADRACKLIMIT = 64$ retransmission attempts). Moreover, tuning the transmission parameters based on the best RSSI of the 20 received packets does not lead to the optimal choice. Therefore, different ADR enhancements have been proposed in [5], [11]–[13]. However, in those proposals, the server-side algorithm can only decrease the SF even in some cases (due to mobility) it should increase the SF to have a better coverage (Low acceptable RSSI range).

In a previous work [2], we have proposed an Enhanced ADR (E-ADR) adapted to node mobility, which consists in fine-tuning the configuration by both decreasing or increasing the SF according to the next node position definition based on a predefined mobility model. Unlike Basic ADR and the other proposed enhancements in the literature that are reactive algorithms, E-ADR is a proactive algorithm characterised by a quick re-adaptivity to RSSI changes. In E-ADR, each node specifies its mobility pattern when joining the server by indicating its application ID in the join request message. Then, the node transmits n packets using $mode1$ in Table I. The server uses the trilateration method to calculate the node position for each received packet. Then, it calculates the average variation $Avg_{variation}$ (displacement) of the node [2].

Based on this average variation, the server estimates the n next positions based on the Linear regression algorithm. When the position is estimated, the server calculates the distance separating the node and the gateways and deduces the estimated RSSI [14].

To decide which configuration the node will be allocated, the server calculates a transition rate $R_{a \to b}$, where a is the old mode and b is the next estimated mode, according to equation (1). $R_{a \to b}$ defines how much $RSSI_{estim}$ belongs to the interval $[B_i, B_s]$ of each configuration in $\{mode1, ...mode10\}$ different to mode a (See Table I).

$$R_{a \to b} = \frac{1/2 \times |B_s(b) - B_i(a)|}{|B_s(b) - RSSI_{(estim)}| + |B_i(b) - RSSI_{estim}|} \quad (1)$$

The greatest element of the vector corresponds to the new configuration mode.

Our previous experimental tests [2] have shown that E-ADR outperforms Basic ADR and the other enhancements ([5], [11]–[13]) thanks to its RSSI-based proactive adjustment strategy. However, this E-ADR version does not cope with applications where nodes travelling through unknown trajectories. So, in what follows we propose an E-ADR extension to include unknown random mobility pattern. As a motivating example, we will consider the cattle monitoring application in a smart farm [3].

III. VHMM(2,1)-BASED E-ADR

Predicting trajectories of moving objects has recently received increasing attention. In [15], the authors presented the hidden markov model (HMM) for the reservation of resources and for the choice of access points in the case of cellular mobile networks. HMM allows the mobile node to learn the environment and update the information itself, thus improving performance. The authors in [16] compared four different families of location predictors that were tested on a set of data collected by Dart-mouth College from Wi-Fi users between April 2001 and March 2003. From the obtained results, the authors concede that the more complex predictors (such as LZ (Lempe and Ziv algorithm) [17], PPM (Prediction by Partial Matching) [18] and SPM (Sampled Pattern Matching) [19]) are not necessarily the more accurate than HMM predictors. They also established that HMM predictors beyond the second-order are less accurate.

In our work, we are interested in using HMM [15] for predicting the position of a mobile node by combining second-order HMM (HMM-2) with first-order HMM (HMM-1). The
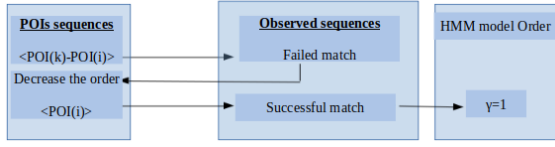
Fig. 1: VHMM(2,1) prediction model

HMM order (γ) defines on how many historical POIs the next estimated POI depends. For example, unlike HMM-1, where only the current state POI(t) determines the next $POI_{(t+1)}$, HMM-2 depends on the current $POI_{(t)}$ and the previous $POI_{(t-1)}$.

The principle idea of our model is that it switches between two HMM orders ($\gamma = 1, 2$) to predict the next node position or point of interest (POI), on which the prediction of the next configuration will be based. In fact, as depicted in Fig. 1, our model named VHMM(2,1) acts primarily as an HMM-2. This latter switches to an HMM-1 when its outcome is an empty set (Failed match) in order to avoid a random prediction.

For our model, we consider $l = 8$ possible states (POIs): Direct (D), Back (A), Right (R), Left (L), Left Lean back (AL), Left Lean forward (PL), Right Lean back (AR), Right Lean forward (PR) ($l = \{D, A, R, L, AL, PL, AR, PR\}$) .

The relevant definitions of VHMM(2,1) are as follows.

Definition 1:
Variable HMM-(2,1) uses $\lambda = (\pi, A_1, A_2, B_1, B_2)$ to describe a model. Supposing that POI{} (POI $\in l$) represents a sequence state of node positions and O{} (O $\in l$) represents the observation state sequence. VHMM-(2,1) is characterised by the following notations:

-π = the probability of the initial state of trajectory, which is known in our case.

-$A_1 = \{A_{t,(t+1)}\}$: $A_{t,(t+1)}$=P($POI_{(t+1)}$| $POI_{(t)}$) is the probability that there is a 1-order sequence: it means that there is $POI_{(t)}$ followed by $POI_{(t+1)}$ in a training (observation) sequence.

-$A_2 = \{A_{(t-1)t,(t+1)}\}$: $A_{(t-1)t,(t+1)}$=P($POI_{(t+1)}$| $POI_{(t)}$,$POI_{(t-1)}$) is the probability that there is a 2-order sequence: it means that there is a sequence $POI_{(t-1)}$ →$POI_{(t)}$ followed by a $POI_{(t+1)}$ in a training (observation) sequence.

-$B_1 = \{B_{t,(t+1)}\}$: $B_{t,(t+1)}$=P($O_{(t+1)_{(l)}}$| $POI_{(t)}$) is the probability of occurrence that defines how many times a $POI_{(t)}$ is followed by each $POI_{(t+1)_{(l)}}$, where l is one of the possible POIs ($l \in \{D, A, R, L, AL, PL, AR, PR\}$).

-$B_2 = \{B_{(t-1)t,(t+1)}\}$: $B_{(t-1)t,(t+1)}$=P($O_{(t+1)_{(l)}}$| $POI_{(t)}$, $POI_{(t-1)}$) is the probability of occurrence that defines how many times a sequence $POI_{(t-1)} \to POI_{(t)}$ is followed by each $POI_{(t+1)_{(l)}}$, where l is one of the possible POIs ($l \in \{D, A, R, L, AL, PL, AR, PR\}$).

The training trajectories of the node's movement in the observation bloc (Fig. 1) are divided into groups where each group is composed of 3 POIs in case of HMM-2 and 2 POIs

in case of HMM-1. Then, we define the probability that a $POI_{(t+1)_{(l)}}$ (l is the POI number) is the succession of the sequence $POI_{(t-1)} \to POI_{(t)}$ in the case of HMM-2 and the succession of only $POI_{(t)}$ in the case of HMM-1.

In case the probability $P(POI_{(t+1)_{(l)}})$ presented in **Definition 2** corresponds to an empty set ($\emptyset$) - meaning that there is no sequence corresponding to $[POI_{(t-1)} \to POI_{(t)} \to P(POI_{(t+1)_{(l)}})]$ in the training sets-, the server switches to the HMM-1 model and will be based on the probability of $P(POI_{(t+1)_{(l)}})$ presented in **Definition 3**.

Definition 2: The probability that $(POI_{(t+1)_{(l)}})$ is the succession of $POI_{(t-1)} \to POI_{(t)}$ (γ=2) is:

$$P(POI_{(t+1)_{(l)}}) = \frac{B_{(t-1)t,(t+1)}}{A_{(t-1)t,(t+1)}} = \frac{P(O_{(t+1)_{(l)}}|POI_{(t-1)},POI_{(t)})}{P(POI_{(t+1)}|POI_{(t-1)},POI_{(t)})}$$

Definition 3: The probability that $(POI_{(t+1)_{(l)}})$ is the succession of $POI_{(t)}$ (γ=1) is:

$$P(POI_{(t+1)_{(l)}}) = \frac{B_{t,(t+1)}}{A_{t,(t+1)}} = \frac{P(O_{(t+1)_{(l)}}|POI_{(t)})}{P(POI_{(t+1)}|POI_{(t)})}$$

The greatest probability $P(POI_{(t+1)_{(l)}})$ will be considered and the corresponding l ($l \in \{D, A, R, L, AL, PL, AR, PR\}$) will be the predicted POI. Based on this POI, the server estimates the next configuration mode using (1). Because the node travels an unknown trajectory, we propose to update the configuration after each $n = 1$ received packet.

IV. EXPERIMENTAL EVALUATION

To evaluate our proposed VHMM-based E-ADR, we have conducted several experiments using Waspmote SX1272 [7], STM32 nodes [20] and gateways considering a smart farm application. In Waspmote SX1272, there are 10 predefined modes (Table I) that will be used in our experimental tests. It is however worth noting that we can choose any parameter combinations without being constrained to only the 10 predefined modes in Table I.

A. Smart farm: A motivating application

As a motivating domain, in this paper, we emulate a smart farm scenario and we consider a cattle monitoring application. Let's consider a LoRaWAN network presented in Fig. 2. It is composed of 5 sensors carried by 5 animals and dedicated for monitoring and tracking, and 3 gateways (G_a, G_b and G_c). We suppose that these animals are following random trajectories in an open field S: S=([X_{min}, X_{max}] = [$-120m, 140m$], [Y_{min}, Y_{max}] = [$0, 140m$]). In order to emulate these animals, nodes are manually handled and moved at an average speed of $9m/min$.

We suppose that 3 sensors (D_2, D_3 and D_4) are sending 21 packets, D_1 is sending 36 packets and D_5 28 packets. Each packet contains 25Bytes and all nodes send packets every 2 minutes ($P_{sending(i)}$). As presented in Fig. 2, we consider the previously presented $L = 8$ possible states (POIs) for our VHMM model.
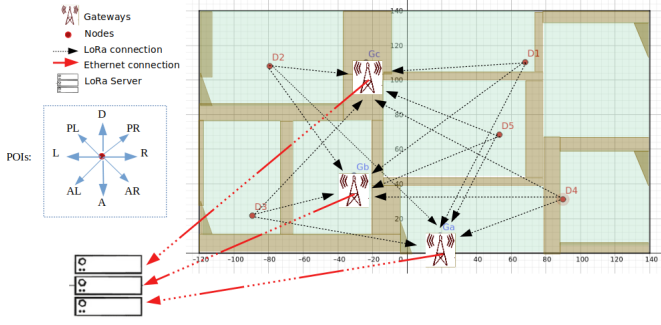
Fig. 2: Test scenario

TABLE II: Prediction accuracy

Node	VHMM(2,1)	HMM-1	HMM-2	HMM-3
D_1	86.08%	54.87%	69.71%	42.25%
D_2	89.14%	57.12%	71.48%	47.51%
D_3	88.52%	55.58%	65.89%	44.47%
D_4	88.09%	52.38%	70.35%	48.02%
D_5	87.15%	54.24%	67.19%	47.87%

B. Measurements

The first phase involved the recording of the trajectories and the directions of cattle over time in order to build a training set consisting of 90 trajectories for all nodes. The trajectories in the training phase are divided into groups of 3 POIs for HMM-2 model ($POI_{(t-1)} \rightarrow POI_{(t)} \rightarrow POI_{(t+1)}$) and 2 POIs for HMM-1 model ($POI_{(t)} \rightarrow POI_{(t+1)}$).

During the test phase, the reported results represent the average of 5 tests for each node. We will predict the next POI ($POI_{(t+1)}$) of each moving node based on the VHMM(2,1) model and the trajectory set experienced during the training phase.

In our experiments, we define $\alpha_{i,j}$ in (2) as the maximum number of allowed retransmissions of the ith packet before the next scheduled packet $(i+1)$ should be sent by D_j. In (2), $T_{delay}(i,j)$ corresponds to the transmission delay of packet (i) sent by D_j at instant T_i and represents the amount of time required to receive successfully the ith packet (including retransmissions and the reception windows). $T_{delay}(i,j)$ can very depending on the used configuration mode and the number of performed retransmissions. $\alpha_{i,j}$ depends also on the sending interval $P_{sending}(j)$ (fixed to $1pkt/2min$ in our scenario) allowing retransmissions to take place only during $[T_i, T_{i+1}]$ (which is an interval of $2min$, in our case).

$$\alpha_{i,j} = \frac{P_{sending}(j)}{T_{delay}(i,j)} \quad (2)$$

During these $\alpha_{i,j}$ retransmissions, if the node does not receive an acknowledgement after a threshold m ($m < \alpha_{i,j}$), it will increment its SF to increase the coverage area and by thus the probability to reach the gateway. In [?], the threshold m is set to 64 by default. In addition, during our experiments, if a packet i sent at T_i is still not acknowledged at instant T_{i+1} (after $\alpha_{i,j}$ retransmission attempts), it will be considered as "lost" because of bad SF allocation. Moreover, when D_j exceeds the duty cycle limitation of 36s per hour due to a high number of retransmissions, the non-transmitted packet will be discarded.

During the experiments, the $n = 2$ first transmissions are established using $Mode1$ to guarantee the history of 2 POIs to perform the HMM-2. Moreover, $mode1$ is chosen to ensure to reach the gateways. Then, each process assigns a parameter

combination according to the used strategy. The configuration will be updated after every $n = 1$.

During the $n = 2$ first transmissions, we recorded all the RSSI values. For D_1, for example, we recorded an RSSI values of $-112.62dB$ and $-115.7dB$ From these RSSI values, the server allocates the configuration mode to be used for the next transmission. According to the received RSSI values, using Basic ADR, the server will allocate $Mode10$ for D_1, for example. However, using VHMM-based E-ADR, based on the next predicted POI for the configuration allocation, the server will allocate $mode8$ for D_1 in order to ensure the needed radio coverage range.

Firstly, we compare the VHMM(2,1) model to (HMM-1, HMM-2, and HMM-3) to validate its prediction efficiency by considering the prediction accuracy criterion defined in (3) as the rate of the number of correctly predicted POIs over the total number of travelled POIs.

$$Accuracy = N_{POI\,correct}/N_{POI\,travelled} \quad (3)$$

Hereafter, the VHMM-based E-ADR is compared to Basic ADR in terms of PLR and energy consumption. In addition, as retransmission has a direct impact on PLR and energy consumption, we will also evaluate the impact of the retransmission threshold m.

C. Evaluation of the prediction model

In this section, we evaluate the prediction accuracy of VHMM(2,1). We have firstly tested multiple random trajectories using (HMM-1, HMM-2, HMM-3 and VHMM(2,1)) models.

Fig. 3 presents the obtained predicted trajectories using each model for one of these random trajectories (using D_2). According to Fig. 3, the VHMM(2,1) model results are better than those obtained using the HMM-1, HMM-2, and HMM-3 models. We note that among the 3 models, HMM-2 is positioned after VHMM(2,1), which is closer to the real covered pattern compared to HMM-1 and HMM-3.

We have registered the accuracy ratio for each tested trajectory. Table II presents the average accuracy prediction using the different HMM models for these tests. The POI is considered to be correctly predicted after checking the real covered trajectory.

In fact, HMM-1 is based only on the current POI to predict the next one, giving a less accurate prediction compared to VHMM(2,1), or HMM-2 which are based on 2 historical POIs to predict the next position (Table II). Furthermore, the HMM-3 model is the less efficient compared to all other models since it is based on 3 historical POIs requiring more learning in
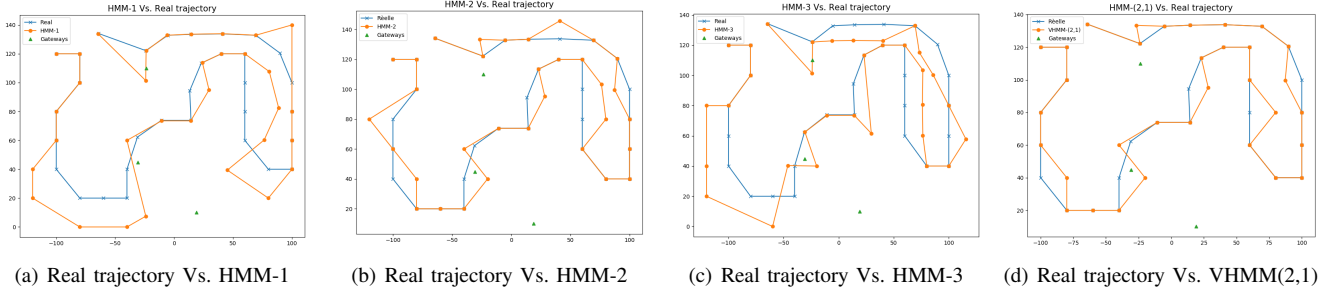
(a) Real trajectory Vs. HMM-1 (b) Real trajectory Vs. HMM-2 (c) Real trajectory Vs. HMM-3 (d) Real trajectory Vs. VHMM(2,1)

Fig. 3: Real trajectory Vs. HMM-1 Vs. HMM-2 Vs. HMM-3 Vs. VHMM(2,1)

TABLE III: Average PLR of D_j

Node	Basic ADR	VHMM-based E-ADR
D_1	41.66%	5.55%
D_2	42.85%	14.28%
D_3	52.38%	0%
D_4	38.09%	0%
D_5	42.85%	7.14%

Fig. 4: PLR evolution for D_2

the training phase. Moreover, the HMM predictors beyond the second-order are in general less accurate [16].

In what follows, we evaluate our VHMM-based E-ADR proposal and compare it to Basic ADR in terms of PLR and energy consumption. To fairly compare them by focusing on the algorithmic efficiency of the strategy without the influence of the algorithm execution frequency, we execute the ADR update every $n = 1$ received packet instead of every $n = 20$ packets for Basic ADR.

D. PLR evaluation

We define PLR as the number of the lost packets (due to bad SF or a lot of interference) and non-transmitted packets (due to cycle limitation) over the total planned number of packets that a node D_j should send during the experiment. Table III presents the average PLRs of 5 measurements for each node D_j using the VHMM-based E-ADR and Basic ADR.

We observe that the nodes experience a high PLR using Basic ADR. In fact, as the server-side in Basic ADR is responsible of only the SF decrease (only the node-side can increase it after a $m = 64$ retransmissions), even if the RSSI value indicates the need of higher SF, the server can not respond to the node requirement inducing by thus an important PLR.

Because of high number of retransmissions, all nodes suffer from a PLR due to duty cycle exceeding in addition to the bad SF allocation. So, to better understand when and how the packets are lost (bad SF) or non-transmitted (duty cycle exceeding), Fig. 4 traces the PLR evolution in time of D_2, by defining separately the number of lost packets (bad SF) over the number of packets already transmitted (blue curves) and the number of non-transmitted packets (exceeding the duty cycle limitation) over the number of the already transmitted ones (orange curves).

We notice in Fig. 4 that during the transmission of the first packets (packets from 1 to 14), the duty cycle is still valid and
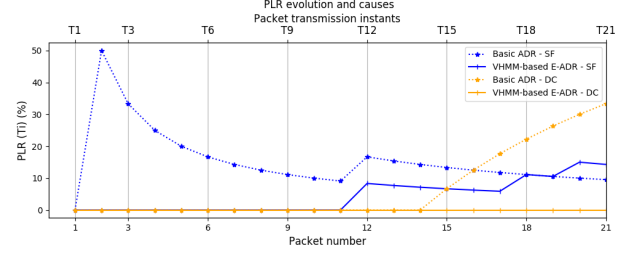
loss is rather due to the bad choice of SF (blue curve). In fact, when nodes are close to the gateways and begin to move away from them, high RSSIs are in the history list, resulting in the attribution of low SFs, inducing an under-optimal allocation, so, a high number of loss. Thus, the retransmission of these packets on the one hand and the sending of the rest of the packets on the other hand will cause the duty cycle to be exceeded and therefore the increase in packet loss (orange curve).

However, using the VHMM-based E-ADR, the packet loss due to bad SF is decreased thanks to the pro-activity of the strategy based on the prediction of the future node locations, inducing less retransmissions and therefore the risk of exceeding the duty cycle.

So, although retransmissions give packets better chance to be successfully received, this does not necessarily mean low PLR, since these retransmissions lead to duty cycle exceeding inducing high PLR because of the not transmitted (discarded) packets.

The VHMM-based E-ADR model allows an optimal configuration allocation allowing to reduce the PLR. In some cases (D_1, D_2, and D_5 for example), nodes may loss a few packets because of the inaccurate POI prediction.

E. Energy consumption evaluation

Retransmission in LoRaWAN increases ToA and so the energy consumption which is a crucial factor for IoT devices. In the following, we will evaluate the energy consumption that is calculated according to [21] for Basic ADR and VHMM-based E-ADR.

Table IV presents the average energy consumption per received packet of each node D_j using both Basic ADR and

TABLE IV: Average energy per received packet (J)

Node	Basic ADR	VHMM-based E-ADR
D_1	0.97	0.341
D_2	1.7	0.73
D_3	2.04	0.26
D_4	1.56	0.26
D_5	1.27	0.34

TABLE V: m impact on energy and PLR

Impact of m on energy (J)				
Node	Basic ADR		VHMM-based E-ADR	
	$m = 8$	$m = 64$	$m = 8$	$m = 64$
D_1	0.63	0.97	0.28	0.341
D_2	0.73	1.7	0.58	0.73
D_3	1.05	2.04	0.26	0.26
D_4	0.75	1.56	0.26	0.26
D_5	0.94	1.27	0.29	0.34
Impact of m on PLR				
Node	Basic ADR		VHMM-based E-ADR	
	$m = 8$	$m = 64$	$m = 8$	$m = 64$
D_1	11.11%	41.66%	0%	5.55%
D_2	14.28%	42.85%	0%	14.28%
D_3	19.04%	52.38%	0%	0%
D_4	4.7%	38.09%	0%	0%
D_5	25%	42.85%	0%	7.14%

VHMM-based E-ADR. These values are defined as: the total consumed energy over the number of received packets.

By allocating a bad SF, nodes using Basic ADR are faced a large number of retransmissions and therefore an increase in energy consumption.

F. Impact of retransmission threshold (m)

As we have noticed, retransmissions may sometimes decrease the PLR at the expense of increasing energy consumption in transitive network condition changes. However, with $m = 64$ by default, a high number of retransmissions may also lead the node to exceed its authorised duty cycle (increasing PLR) even before the node can reach a suitable SF by linearly incrementing it every m retransmissions. Table V presents the impact of m on respectively the average energy per received packet and the PLR by varying m ($m = \{8, 64\}$).

The obtained results in Table V show that a low m value ($m = 8$) allows more chance to reach quickly the optimal SF, as the node increments its SF every m retransmissions, avoiding a high number of retransmissions, a duty cycle exceeding, an important PLR and high energy consumption. However, even if with a small retransmission threshold (eg., $m = 8$) the Basic ADR still has higher power consumption compared to VHMM-based E-ADR, since it still has recourse to retransmissions. Furthermore, using VHMM-based E-ADR, nodes did not have to re-transmit packets thanks to the optimal allocation using the prediction of the next POI based on the VHMM(2,1) model. So, VHMM-based E-ADR offers a good trade-off between energy consumption and PLR.

V. CONCLUSION

ADR may be used to re-adapt the node configuration in LoRaWAN networks, but it is facing to low adaptation speed and low performance issues in case of mobility. VHMM-based E-ADR solves those issues in case of mobility with undefined trajectories. We have shown through our experimental tests that our proposed strategy based on the prediction of the device displacement outperforms Basic ADR by offering a fair trade-off between PLR and energy consumption. Moreover, by reducing PLR, the number of retransmissions is minimised, and thus the probability to exceed the duty cycle limitation. Our experiments have shown that our prediction model "VHMM(2,1)" presents 87% accuracy. In Future work, we will be interested to improve that by increasing the number of experimented devices and by exploring other ML techniques. Furthermore, we will study the scalability of our proposed mechanism in a large network.

REFERENCES

[1] L. Alliance, "LoRaWAN Specification," v1.0.2, 2016.
[2] N. Benkahla, H. Tounsi, Y.Q. Song, and M. Frikha. "Enhanced ADR for LoRaWAN networks with mobility," In: *15th IWCMC*, Morroco, 2019.
[3] Semtech. "Revolutionizing Smart Agriculture Using Semtech's LoRa Technology," Semtech White paper, October 2017
[4] A. Augustin and al., "A study of LoRa: Long range low power networks for the Internet of Things," Sensors, vol. 16, no. 9, 2016.
[5] M.C. Bor, U. Roedig, "LoRa Transmission Parameter Selection," In the *13th DCOSS*, Ottawa, Canada, 5–7 June 2017, pp. 27–34.
[6] LoRa Alliance Technical committee, LoRaWAN - Regional Parameters, 1.0, LoRa Alliance, 45 pp, 2016.
[7] Libelium comunicaciones Distribuidas S.L., "Waspmote LoRa 868 MHz 915 MHz SX1272 networking guide, " Rev 4.2, November 2015.
[8] O. Alvear, J. H-Tapia, C. Calafate, E. H-Orallo, J-C Cano, and P. Manzoni, "Assessing the Impact of Mobility on LoRa Communications". In: *InterIoT/SaSeIoT'17*, Spain, 2017.
[9] R. S-Cabrera, A. Pachon, and J-M. Madrid. "Proof of Concept of an IoT-Based Public Vehicle Tracking System, Using LoRa (Long Range) and Intelligent Transportation System (ITS) Services," In: *JCNC*. vol 19, 2019
[10] K. Kousias, G. Caso, O. Alay, and F. Lemic, "Empirical Analysis of LoRaWAN Adaptive Data rate for Mobile Internet of Things Applications," In: *Wireless of the Students Workshop*, Mexico, 2019.
[11] M. Slabicki, G. Premsankar, and M. Di Francesco. "Adaptive configuration of loRa networks for dense IoT deployments," In: *NOMS IEEE/IFIP*. Taiwan, 2018.
[12] F. Cuomo, Ma. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," In: *13th IEEE WiMob*, Italy, 2017.
[13] D.-Y. Kim, S. Kim, H. Hassan, J.H. Park, "Adaptive data rate control in low power wide area networks for long range IoT services, " Journal Comput. Sci.2017,22, 171–178.
[14] E. Goldoni, L. Prando, A. Vizziello, P. Savazzi, P. Gamba. (2019). "Experimental data set analysis of RSSI-based indoor and outdoor localization in LoRa networks". In: Internet Technology Letter.
[15] H. Zhang and L. Dai, "Mobility Prediction : A Survey on State-of-the-Art Schemes and Future Applications,". In *IEEE Access*, vol. 7, 2019, pp. 802-822.
[16] L. Song, D. Kotz, R. Jain, and X. He. "Evaluating next-cell predictors with extensive wi-fi mobility data," IEEE Transactions on Mobile Computing, Vol. 5, CA, USA, 2006, pp. 1633-1649.
[17] A. Bhattacharya and S.K. LeZi. "An information-theoretic approach to track mobile users in PCS networks," ACM/Kluwer Wireless Networks, May 2002, pp. 121–135.
[18] T.C. Bell, J.G. Cleary, and I.H. Witten. "Text Compression," Prentice Hall, 1990.
[19] P. Jacquet, W. Szpankowski, and I. Apostol. "An universal predictor based on pattern matching,preliminary results," Mathematics and Computer Science, chapter 7, Birkhauser, 2000, pp. 75–85.
[20] Semtech Corporation, Discovery kit for LoRaWAN, Sigfox, and LPWAN protocols with STM32L0 B-L072Z-LRWAN1, 2019
[21] T. Bouguera, J-F. Diouris, J-J. Chaillout, R. Jaouadi, and G.Andrieux, "Energy Consumption Model for SensorNodes Based on LoRa and LoRaWAN," MDPI Sensors, 2018.