Upper-Confidence Bound for Channel Selection in LPWA Networks with Retransmissions

Rémi Bonnefoi[1], Lilian Besson[1], Julio Manco-Vasquez[1], and Christophe Moy[2]

[1] IETR / CentraleSupélec Campus de Rennes, F-35510 Cesson-Sévigné, France,
{Remi.Bonnefoi,Lilian.Besson,JulioCesar.MancoVasquez}@CentraleSupelec.fr
[2] Univ Rennes, CNRS, IETR - UMR 6164, F-35000, Rennes, France
Christophe.Moy@Univ-Rennes1.fr

***Abstract*—In this paper, we propose and evaluate different learning strategies based on Multi-Arm Bandit (MAB) algorithms. They allow Internet of Things (IoT) devices to improve their access to the network and their autonomy, while taking into account the impact of encountered radio collisions. For that end, several heuristics employing Upper-Confident Bound (UCB) algorithms are examined, to explore the contextual information provided by the number of retransmissions. Our results show that approaches based on UCB obtain a significant improvement in terms of successful transmission probabilities. Furthermore, it also reveals that a pure UCB channel access is as efficient as more sophisticated learning strategies.**

***Index Terms*—Low Power Wide Area, Multi-Armed Bandits, Upper-Confident Bound, retransmissions, Internet of Things.**

I. Introduction

Nowadays, the Internet of Things (IoT) and in particular the Low Power Wide Area (LPWA) technology is considered a main driver for a vast variety of application that will support the communications among a large number of devices. In fact, network operators are starting to deploy Machine to Machine (M2M) solutions using LPWA networking technologies [1]. For instance, LoRaWAN and SigFox technologies have been most adopted in the monitoring of large scale systems (*e.g.*, smart cities, metering), where a large number of devices compete for the transmission of their packets in the unlicensed Industrial, Scientific and Medical (ISM) bands.

Nevertheless, this demand to fit a growing number of energy-limited end-devices requires the development of contention-based protocol more tailored for LPWAN technologies. Thus, novel access mechanisms considering collision-avoidance methods need to be addressed to avoid degrading the network performance in these unlicensed bands. In fact, the number of packet collisions increases as more devices without coordination share the same band. Hence, an important concern in the Medium Access (MAC) design is to reduce the Packet Loss Ratio (PLR) due to the interference caused by the collisions among the devices.

This publication is supported by the French National Research Agency (ANR), under the projects SOGREEN and EPHYL (grants *N ANR-14-CE28-0025-02* and *N ANR-16-CE25-0002-03*), by Région Bretagne, France, by École Normale Supérieure de Paris-Saclay. by European Union, through the European Regional Development Fund (ERDF), and by Ministry of Higher Education and Research, Brittany and Rennes Mtropole, through the CPER Project *SOPHIE / STIC & Ondes*.

In this regard, in the context of Cognitive Radio [2], [3], Multi-Arm Bandit (MAB) algorithms [4], [5], [6] have been recently proposed as a potential solution for channel access in LPWA networks [7], [8], [9]. For instance in [9], the impact of non-stationarity on the network performance using MAB algorithms is studied. In this work, low-cost algorithms following two well-known approaches, such as the Upper-Confidence Bound (UCB) [4], [5], and the Thompson Sampling (TS) algorithms [10] have reported encouraging results. Other recent directions include theoretical analysis [11], [12], and realistic empirical simulations [13], [14], of the application of MAB algorithms for slotted wireless protocols in a decentralized manner, or applications to multi-hoping networks [15], [16]. None of the above mentioned articles discusses in detail the impact of retransmissions on the performance of MAB learning algorithms as we do in this paper.

The aim of this paper is to assess the performance of MAB algorithms [6] for channel selection in LPWA networks, while taking into account the impact of retransmissions on the network performance. For this reason, several decision making strategies are applied after a first retransmission (*i.e.*, when a collision occurs). Proposed approach employs contextual information provided by the number of retransmissions, and implemented at each device, so that no coordination among them is needed. Moreover, our UCB-based heuristics show low complexity making them suitable for being embedded in LPWA devices.

The contributions of this paper are summarized as follows:

- Firstly, we provide a close form approximation of the radio collision probability after a first retransmission. By doing this, we highlight the need to develop a learning approach for channel selection upon collision.
- Secondly, several heuristics are proposed to cope with retransmissions.
- Lastly, we conduct simulations in order to compare the performance of the proposed heuristics with a naive uniform random approach, and a UCB strategy (*i.e.*, without any learning for the retransmissions).

The rest of the paper is organized as follows. First the system model is introduced in Section II. A formal description of the MAB learning algorithms is given in Section III, and

our motivations are exposed in Section IV. The proposed UCB-based heuristics are presented in Section V, while the corresponding numerical results are shown in Section VI. Finally, some conclusions are drawn in Section VII.

II. System model

A. LPWA Network

We consider in this paper an LPWA network composed of a gateway and a large number of end-devices that regularly send short data packets, where K channels ($K > 1$) are available for the transmission of their packets.

We assume that this network is constituted by two types of devices: on the one hand, we have *static* devices that operate in one channel[1] in order to communicate with the gateway. On the other hand, there are IoT devices, that possess the additional advantage of being able to select any of the K available channels to perform their transmissions.

Regardless the type of devices, each of them follows a slotted ALOHA protocol [17], and has a probability $p > 0$ to transmit a packet in a time slot. We make the hypothesis that the transmission is successful if the channel is available, otherwise upon radio collision, these devices will attempt to retransmit their packet up-to M times, with $M \in \mathbb{N}$. Note that, every retransmission is carried out after a random back-off time, uniformly distributed in $[\![0, m-1]\!]$, where $m \in \mathbb{N}, m > 0$ is the length of the back-off interval.

B. Model of our IoT devices

The aforementioned contention process can be described by a Markov chain model [18] similar to the one presented in [19], as it is depicted in Fig. 1. A device containing a packet for transmission goes from an idle state to a transmission state, while considering retransmissions due to different collision probabilities, *i.e.*, $\{p_c, p_{c1}, \ldots, p_{M-2}\}$, at each M back-off stage. At each time slot, a transition from an idle state to a transmission state (denoted as `Trans.`) occurs if a packet transmission is required, while waiting states (denoted as `Wait`), correspond to a m back-off interval.

A device aims to select a channel with the highest probability of a successful transmission, for which it resorts to a reinforcement learning approach. It is formulated as MAB problem, where each channel (also called arms) is viewed as a gambling machine (bandit), and each bandit has a *reward*. Then, at every trial, a device chooses a channel that maximizes the sum of the collected rewards. These *rewards* are the *acknowledgment* (*Ack*) signals received after transmitting packets to the gateway. In this way, a successful transmission is considered when an acknowledgment is received, and a learning approach is employed to select the best channel.

We address the problem of channel selection taking into account the described Markov model for the retransmissions of end-devices. It motivates our present work for which we consider the retransmissions in the analysis of MAB algorithms.

[1] Note that, for unlicensed bands, this definition also encompasses any device following a different standard or trying to establish communication with gateways of other networks.

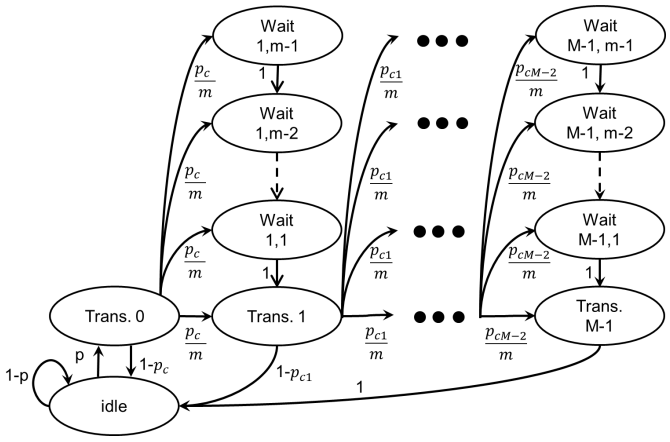

Fig. 1. All devices in the network follow the same Markov behavior.

III. Motivations for the proposed approach

When a device experiments a collision, it goes in a back-off state to retransmit the same packet on a channel. If all devices remain in the same channel for retransmissions, it could result in a sequence of successive collisions with the same packets' devices that previously collided. Thus, it seems interesting to consider in the decision making policy the possibility for a device to retransmit in a different channel. One of our motivations to develop new MAB algorithms for our problem is this option of using a different communication channels between the first transmission and the next retransmissions.

By considering this possibility, the device will have to learn more, thus, we expect that the learning time to be longer, but it could be possible that the final performance gain (*i.e.*, in terms of network performance) increases too. The next Section VI presents analysis to check this performance gain, for various heuristics based on the UCB algorithm.

Here after, we start by presenting a mathematical derivation that backups this idea. To do so, we study the collision probabilities considering the Markov process depicted in Fig. 1, and foresee the impact of addressing bandit strategies, as well as setting guidelines for the design of heuristic approaches.

A. Probability of collision at a second transmission slot

As it is well known, having a collision during an access time can be overcome by a retransmission procedure (this can take several retransmission attempts). What interest us here, is to obtain a mathematical approximation of the collision probability at the second transmission slot p_{c1}, as a function of the first collision probability p_c.

We consider two hypotheses $\mathcal{H}_1$ and $\mathcal{H}_2$ defined as,

- $\mathcal{H}_1$: The probability p_{c1} is composed by the sum of two probabilities: i) the probability of colliding consecutively twice, *i.e.*, the devices that collide at a given time slot and collide again when retransmitting their packets, and ii) the probability of collision related with devices that attempt to transmit in this channel for first time. In addition, we suppose that the number of devices that attempt to

retransmit is small in comparison to the total number of devices sharing the same channel.

- $\mathcal{H}_2$: The total number of the back-off stages at time t is constant, and it is assumed to be large enough to consider that no device will ever be in the last failure state (this case is the one on the right side in Figure 1), after M successive failed retransmissions.

Considering one device and a channel, we denote x_t^i the probability that it is transmitting a packet for the $i+1$ time in a given time slot t (with $i \in [\![0, M-1]\!]$), and let $x_t = \sum_{i=0}^{M-1} x_t^i$ be the probability that it transmits a packet. We consider N active devices following the same policy.

We assume to be in the steady state [18], in our Markov chain model depicted in Figure 1, and thus the probabilities no longer depend on the slot number t (*i.e.*, $\forall t, x_t = x$). Therefore, the probability that this device has a collision at the first transmission is p_c, and has the following expression

$$p_c = 1 - (1 - x_t)^{N-1} \iff x_t = 1 - (1 - p_c)^{\frac{1}{N-1}}. \quad (1)$$

Moreover, from (1) we define the probability $p_{cp}(n)$ that involves the collision of n packets sent by each IoT device (for any $1 \leq n \leq N-1$), during the first transmission slot, and is defined by the following equation

$$p_{cp}(n) = \binom{N-1}{n} x^n (1-x)^{N-1-n}.$$

As explained above, if an IoT device is experiencing a collision at the first transmission, it proceeds for the retransmission of its packet after a random back-off interval. We denote p_{ca} the probability to have a collision with a packet involved in the previous collision. Under assumption $\mathcal{H}_1$, the probability that the same device's packet is experiencing again a collision at the second time slot is

$$p_{c1} = p_{ca} + (1 - p_{ca})\, p_c. \quad (2)$$

If the device had a collision, we consider $p_{bp}(n)$ the probability that it had a collision with *exactly* n packets (for any $1 \leq n \leq N-1$), and that *at least one* of the n devices involved in the previous collision chose the same back-off interval. Hence, under hypothesis $\mathcal{H}_2$, we can relate p_{ca} with $p_{bp}(n)$, and the different probabilities that a device experienced a collision during the first slot and has the same back-off interval for its retransmission is,

$$p_{ca} = \sum_{n=1}^{N-1} p_{bp}(n). \quad (3)$$

Therefore, the expression of p_{ca} is

$$\frac{1}{p_c} \sum_{n=1}^{N-1} \binom{N-1}{n} x^n (1-x)^{N-1-n} \left[1 - \left(1 - \frac{1}{m}\right)^n\right]. \quad (4)$$

Once again under $\mathcal{H}_1$, assuming that the number of devices involved in the first collision is small compared to $N-1$, the first terms of the sum in (4) are predominant. Moreover, for these terms, n is small compared to $N-1$, and so $N-1-n$ can be approximated to $N-1$. Thus it gives,

$$\begin{aligned} p_{ca} =& 1 - \frac{1}{p_c} \sum_{n=1}^{N-1} \binom{N-1}{n} x^n (1-x)^{N-1-n} \left(1 - \frac{1}{m}\right)^n, \\ \simeq& 1 - \frac{(1-x)^{N-1}}{p_c} \sum_{n=1}^{N-1} \binom{N-1}{k} x^n \left(1 - \frac{1}{m}\right)^n. \end{aligned} \quad (5)$$

We use the binomial theorem to compute the sum in (5), and we rewrite the expression of p_{ca} as

$$p_{ca} \simeq \frac{1}{p_c} - \left(\frac{1}{p_c} - 1\right) \left[1 + \left(1 - (1-p_c)^{\frac{1}{N-1}}\right)\left(1 - \frac{1}{m}\right)\right]^{N-1}. \quad (6)$$

Finally, our approximation of p_{c1} can be obtained by inserting (6) in (2).

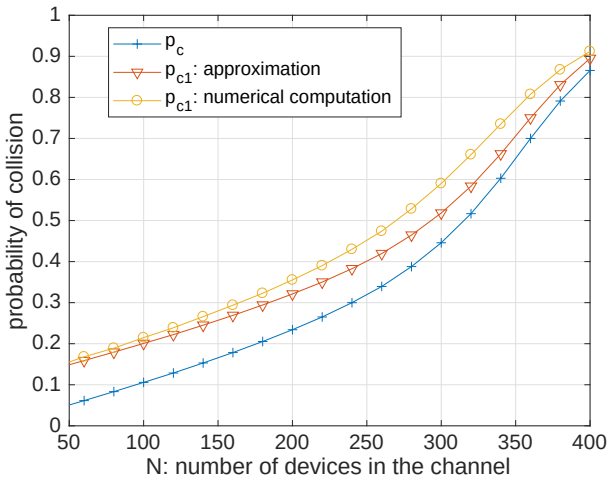

Fig. 2. Our proposed approximation for the probability of collision at the second transmission. It is more precise for smaller values of N.

B. Behaviour analysis of p_c and p_{c1}

In order to assess the proposed approximation, we suppose a unique channel where all the devices follow the same contention Markov process. We simulate an ALOHA protocol with a maximum number of retransmissions $M = 10$, a maximum back-off interval $m = 10$, and a transmission probability $p = 10^{-3}$. In Fig. 2, we show the collision probabilities for different number of devices N (from $N = 50$ up-to $N = 400$), for both p_c and p_{c1}.

From this simulations, we can verify that our approximation is very precise for lower values $p_{c1} \leq 30\%$ (*i.e.*, red and orange curves are quite close). Moreover, a significant gap between p_{c1} and p_c, of up-to 10%, can be observed, which suggests us to resort to MAB algorithms for the channel selection for both the first transmission and next retransmissions.

C. Learning is useful for non-congested networks

It is worth to highlight that, if we write (2) as $p_{c1} = p_c + p_{ca}(1-p_c)$, then it is obvious that p_{c1} is always larger than p_c (as $p_{ca}(1-p_c) > 0$). But for large values of p_c, $p_{ca}(1-p_c) \simeq 0$ so the gap gets small, and for small values of p_c the gap is significant. Moreover, we can verify (*e.g.*, numerically or by differentiating) that the gap decreases when p_c increases (for fixed N and m). This backups mathematically the observation we made from Fig. 2: the smaller the p_c, the larger is the gap between p_c and p_{c1}.

We interpret this fact in two different situations. On the one hand, in a congested network, when devices suffer from a large probability of collision on their first transmission (*i.e.*, p_c is not so small), then $p_{c1} \simeq p_c$ and so devices cannot really hope to reduce their collision probabilities even if the use a different channel for retransmission. On the other hand, if p_c is small enough, *i.e.*, in a network not yet too congested, then our derivation shows that $p_{c1} \gg p_c$, meaning that the possible gain of retransmitting in a different channel that the one used for the first transmission can be large, in terms of collision probability (*e.g.*, up-to 10% in this experimental setting). In other words, when learning can be useful (small p_c), learning to retransmit in a different channel can have a large impact on the global collision rate, thus justifying our approach.

IV. A well-known MAB Algorithm: UCB

Without loss of generality, we have adopted a well-studied stochastic MAB learning algorithm, where the reward distributions are unknown and assumed to be independent and identically distributed (i.i.d). The arms model the channels denoted as $C(t) \in [\![1, K]\!]$, and the players, the dynamic devices, learn the distributions to be able to progressively focus on the best arm, *i.e.*, the arm with largest mean representing the mean availability of a given channel k.

Before presenting our proposed heuristics, we describe a UCB bandit algorithm [4]. It has reported to be efficient, while featuring a low complexity for its implementation. For this reason, it has been employed for IoT applications [9], and we employ this approach to develop our proposals.

A. The UCB algorithm

A first approach is to only use an empirical mean estimator of the rewards in every channel, and select the channel with highest estimated mean at every time step; but this greedy approach is known to fail dramatically [5]. Indeed, with this policy, the selection of arms depends too much on the first draws: if the first transmission in one channel fails and the first one on other channels succeeds, the device will *never* use the first channel again, even if it is the best one (*i.e.*, the most available, in average).

Rather than relying on the empirical mean reward, UCB algorithms instead use a *confidence interval* on the unknown mean μ_k of each arm, which can be viewed as adding a "bonus" exploration to the empirical mean. They follow the "*optimism-in-face-of-uncertainty*" principle: at each step, they play according to the best model, as the statistically best possible arm (*i.e.*, the highest UCB) is selected.

More formally, for one device, let $N_k(t)$ be the number of times the channel k (for $k \in [\![1, K]\!]$) was selected up-to time $t-1$, for $t \geq 0$ for any $t \in \mathbb{N}$,

$$N_k(t) = \sum_{\tau=0}^{t-1} \mathbb{1}(C(\tau) = k), \tag{7}$$

where $\mathbb{1}$ is an indicator function that is equal to 1, if the IoT device chooses, for its τ-th transmission, a channel k, and 0 otherwise. The empirical mean estimator $\widehat{\mu_k}(t)$ of channel k is defined as the mean reward obtained up-to time $t-1$,

$$\widehat{\mu_k}(t) = \frac{1}{N_k(t)} \sum_{\tau=0}^{t-1} r_k(\tau) \mathbb{1}(C(\tau) = k). \tag{8}$$

where $r_k(t)$ is the reward obtained after transmission in channel k at time t (1 for a successful transmission, and 0 otherwise) A *confidence* term $B_k(t)$ is given by [5],

$$B_k(t) = \sqrt{\alpha \log(t)/N_k(t)}, \tag{9}$$

where α refers to an exploration coefficient[2], that we chose equal to $1/2$, as suggested in [20] and as done in previous works [7], [9]. Then, an upper confidence bound in each channel k is defined as

$$U_k(t) = \widehat{\mu_k}(t) + B_k(t). \tag{10}$$

Finally, the transmission channel at time step t is the one maximizing this UCB index $U_k(t)$, as it is the one expected to be the best one at the current time step t,

$$C(t) = \arg \max_{1 \leq k \leq K} U_k(t). \tag{11}$$

The UCB algorithm is implemented independently by each device, and we present it in Algorithm 1. Note that a device using this first approach is only able to select a channel for the first and all the corresponding retransmissions of a packet.

for $t = 0, \ldots, T$ **do**
  Compute for each channel $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$.
  following Eqs. (7), (8), and (9);
  Transmit in channel $C(t) = \arg\max_{1 \leq k \leq K} U_k(t)$;
  Reward $r_{C(t)}(t) = 1$, if *Ack* is received, else 0;
end

Algorithm 1: The UCB algorithm for channel selection.

V. Proposed Heuristics

A device that that implements the UCB algorithm is led to focus is transmissions and retransmissions in the channel which has been identified as the best. As explained in Section III, focusing in one channel increases the collision probability in retransmissions. In this Section, we describe the proposed heuristics for the channel selection in a retransmission. It is carried out taking into account that a device can incorporate

[2] In fact, the larger this coefficient is, the longer the exploration, while the UCB algorithm is proven to be order optimal for $\alpha > 0.5$ [6], and has reported a good performance for lower values of $\alpha > 0$.

a different channel selection strategy while being in a back-off state. Hence, a natural question is to evaluate whether using this additional contextual information can improve the performance of a learning policy.

For that end, all of our heuristics comprise two stages: the first stage is a UCB algorithm employed for the first attempt to transmit, and the second stage is another algorithm used for channel selections for the next retransmissions.

We present below four heuristics for this second stage (short names in "quotes" correspond to the legend on Figures 3, 4).

A. *Uniform random retransmission ("Random")*

In this first proposal, the device uses a random channel selection, following a uniform distribution (in $[\![1, K]\!]$). It is described below in Algorithm 2.

for $t = 0, \dots, T$ **do**
  if *First packet transmission* **then**
    Use first-stage UCB as in Algorithm 1.
  else `// Random retransmission`
    Transmit in channel $C(t) \sim \mathcal{U}(1, \dots, K)$;
  end
end

Algorithm 2: Uniform random retransmission.

B. UCB *for retransmission ("Only* UCB*")*

Instead of applying a random channel selection, another heuristic is to use a second UCB algorithm in the second stage. In other words, we expect that this algorithm is able to learn the best channel to retransmit a packet. It is described in Algorithm 3, and it is still a practical approach, since the storage requirements and time complexity remains linear w.r.t. the number of channels K (*i.e.*, of order $\mathcal{O}(K)$).

Note that, we use the subscript $(^r)$ to denote the variables $\widehat{\mu_k^r}(t)$, $B_k^r(t)$ and $U_k^r(t)$, related to the UCB algorithm employed for the retransmission.

for $t = 0, \dots, T$ **do**
  if *First packet transmission* **then**
    Use first-stage UCB as in Algorithm 1.
  else `// Random retransmission`
    Compute for each channel $U_k^r(t) = \widehat{\mu_k^r}(t) + B_k^r(t)$ following Eqs. (7), (8), and (9);
    Transmit in channel $C^r(t) = \arg\max_{1 \le k \le K} U_k^r(t)$;
    Reward $r_{C^r(t)}^r(t) = 1$, if *Ack* is received, else 0;
  end
end

Algorithm 3: UCB for retransmission.

C. K *different* UCB*s for retransmission ("*K UCB*")*

Another heuristic is to not use the same algorithm no matter where the collision occurred, but to use K different UCB algorithms. Meaning that after a failed first transmission in channel j, the device relies on the k-th algorithm to decide its retransmission. The corresponding algorithm is depicted in Algorithm 4. Each of these algorithms are denoted using the subscript $(^j)$, for $j \in [\![1, K]\!]$.

Although, this approach increases the complexity and store requirements (of order $\mathcal{O}(K^2)$). For our LPWA networks of interest, such as LoRaWAN, the cost of its implementation is still affordable, since a small number of channels is used. For instance, for $K = 4$ channels, the memory to store $K + 1 = 5$ algorithms is of the order of the requirements to storing one.

for $t = 0, \dots, T$ **do** `// At every time step`
  if *First packet transmission* **then**
    Use first-stage UCB as in Algorithm 1.
  else `// Packet retransmission`
    j $\leftarrow$ last channel selected by first-stage UCB;
    Compute for each channel $U_k^j(t) = \widehat{\mu_k}^j(t) + B_k^j(t)$ following Eqs. (7), (8), and (9);
    Transmit in channel $C^j(t) = \arg\max_{1 \le k \le K} U_k^j(t)$;
    Reward $r_{C^j(t)}^j(t) = 1$ if *Ack* is received, else 0;
  end
end

Algorithm 4: K different UCBs for retransmission.

D. *Delayed* UCB *for retransmission ("Delayed* UCB*")*

This last heuristic is a composite of the random retransmission (Algorithm 2) and the UCB retransmission (Algorithm 3) approaches. Instead of starting the second stage UCB directly from the first retransmission, we introduce a fixed delay $\Delta \in \mathbb{N}$, $\Delta \ge 1$, and start to rely on the second stage UCB after Δ transmissions. The selection for the first steps is handled with the random retransmission.

The idea behind this delay is to allow the first stage UCB to start learning the best channel, before starting the second stage UCB (see details in Algorithm 5). The number of transmissions to wait before applying the second algorithm is denoted by Δ, it has to be fixed before-hand.

Note that, we use the subscript $(^d)$ to denote the variables related to the delayed second-stage UCB algorithm.

for $t = 0, \dots, T$ **do** `// At every time step`
  if *First packet transmission* **then**
    Use first-stage UCB as in Algorithm 1.
  else if $t \le \Delta$ **then** `// Random selection`
    Transmit randomly in a channel $C(t) \sim \mathcal{U}(1, \dots, K)$.
  else `// Delayed UCB`
    Compute for each channel $U_k^d(t) = \widehat{\mu_k^d}(t) + B_k^d(t)$ following Eqs. (7), (8), and (9);
    Transmit in channel $C^d(t) = \arg\max_{1 \le k \le K} U_k^d(t)$;
    Reward $r_{C^d(t)}^d(t) = 1$ if *Ack* is received, else 0;
  end
end

Algorithm 5: Delayed UCB for retransmission.

VI. Simulations to compare our heuristics

We simulate our network considering N devices following the contention Markov process described in Section II, and a LoRa standard with $K = 4$ channels. Each device is set to transmit with a fixed probability $p = 10^{-3}$, *i.e.*, a packet about every 20 minutes for time slots of 1 s.

For the evaluation of the proposed heuristics, a total number of $T = 10^6$ time slots is considered, and the results are averaged over 10^3 independent random simulations.

In a first scenario, we consider a total number of $N = 1000$ IoT devices, with a non-uniform repartition of static devices given by $10\%, 30\%, 30\%, 30\%$ for the four channels. In other words, the channels are occupied 10%, 30%, 30%, and 30% of time, and the contention Markov process considered is given by $M = 5$, and $m = 5$. In Fig. 3, we show the successful transmission rate versus the number of slots, for all the proposed heuristics.

A first result is that all the heuristics clearly outperform the non-learning approach that simply use random channel selection for both transmissions and retransmissions (*i.e.*, the **no** UCB curve). The improvement of the heuristics over the non-learning approach is evident, and for every heuristic that use a kind of learning mechanism it can observed a successful transmission rate that increases rapidly (or equivalently an PLR decreasing). Moreover, all of these approaches show a fast convergence making them suitable for the targeted application. It is also worths mentioning that the employment of the same UCB algorithm for retransmissions denoted here as "Only UCB" achieves a better performance, while a "Random" retransmission features a slight degradation. This result can be explained as follows: the loss of performance related to the separation of information for several algorithms is greater than the gain obtained by considering the first transmissions and retransmissions separately.

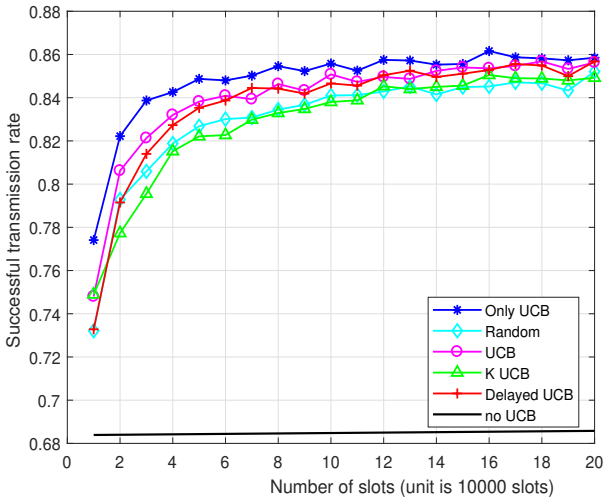

Fig. 3. Comparison among the exposed heuristics for the retransmission: Only UCB, Random, UCB, K UCB, and Delayed UCB. First scenario: learning helps but learning to retransmit smartly is not needed, as we observe that the random retransmission heuristic achieves similar performance than the others.

We also consider in our analysis the case where $M = 5$, and $m = 10$ using ALOHA protocol, a statistic distribution of the devices about $40\%, 30\%, 20\%, 10\%$ for the four channels, and $N = 2000$ IoT devices. The corresponding results are depicted in Fig. 4. In this case the successful transmission rate is degraded compared with achieved results in Fig. 3, this can be explained with the fact that we are considering in our network more devices that increase the collision probability. It is important to highlight,that the "Random" retransmission heuristic shows a poor performance in comparison to the other heuristics, and it can be attributed to the fact that the number of retransmission is increased, and consequently a learning approach is able to take advantage of it. Furthermore, the "Delayed UCB" and the"UCB" heuristics show quite similar results, after converging.

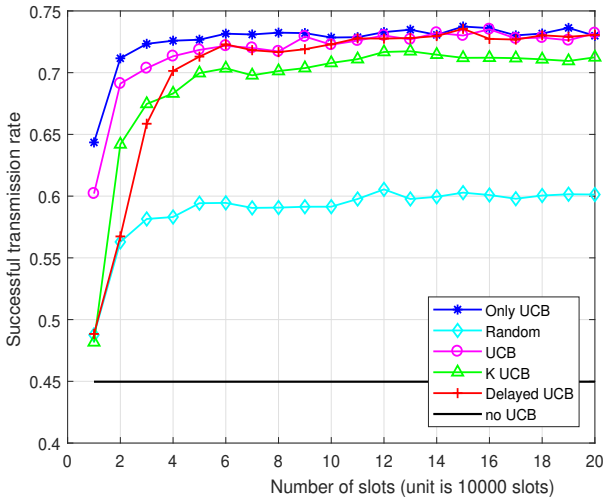

Fig. 4. Second scenario: learning helps a lot (a gain of 30% in terms of collision probability), and learning to retransmit smartly is needed.

The conclusions we can draw from depicted results are twofold. First, MAB learning algorithms are very useful to reduce the collision rate in LPWA networks, a gain of up-to 30% of successful transmission rate is observed after convergence. A second conclusion that can be highlighted is that, using learning mechanisms for retransmissions can be an interesting way to reduce collisions in networks with massive deployments of IoT as this can be checked in Fig. 4, where the random retransmission heuristic is not very advantageous in front of the UCB-based approaches that use learning for channel selection during the retransmission procedure.

VII. Conclusions

In this paper, we presented a retransmission model of LPWA networks based on an ALOHA protocol, slotted both in time and frequency, in which dynamic IoT devices can use machine learning algorithms, to improve their PLR when accessing the network. The main novelty of this model is to address the packet retransmissions upon radio collision, by using a Multi-Armed Bandit framework. We presented and evaluated several learning heuristic that try to learn how to transmit and retransmit in a smarter way, by using the UCB algorithm for channel selection for first transmission, and different proposals based on UCB for the retransmissions upon collisions.

We showed that incorporating learning for the transmission is needed to achieve optimal performance, with significant gain in terms of successful transmission rate in networks with a large number of devices (up-to 30% in the example network). Our empirical simulations show that each of our proposed heuristic outperforms a naive random access scheme. Surprisingly, the main take-away message is that a simple UCB learning approach, that retransmit in the same channel, turns out to perform as well as more complicated heuristics.

Future works

The utility and impact of the proposed approaches for LPWA networks motivates us to address several subjects as future works. Among them, the non-stationarity of the channel occupancy caused by the learning policy employed by the IoT devices. For that end, modifications of MAB algorithms have been proposed, such as Sliding-Window-UCB or Discounted-UCB [21] or more recently M-UCB [22], that nevertheless have not been explored for the targeted problem.

In order to validate our results in a realistic experimental setting and not only with simulations, future works include a hardware implementation of the analyzed models to complete our recent works [23], [24]. A hardware demonstrator could be also benefit to study other settings by removing some hypotheses, for instance by studying a similar model in non-slotted time.

Note on the simulation code

The source code (MATLAB or Octave) used for the simulations and the figures is open-sourced under the MIT License, at `Bitbucket.org/scee_ietr/ucb_smart_retrans`.

REFERENCES

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[2] J. Mitola and G. Q. Maguire, "Cognitive Radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, pp. 13–18, Aug 1999.

[3] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multi-armed Bandit Problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The Non-Stochastic Multi-Armed Bandit Problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[6] S. Bubeck, N. Cesa-Bianchi, *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[7] R. Bonnefoi, C. Moy, and J. Palicot, "Improvement of the LP-WAN AMI backhaul's latency thanks to reinforcement learning algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 34, 2018.

[8] A. Azari and C. Cavdar, "Self-organized Low-power IoT Networks: A Distributed Learning Approach," in *IEEE Globecom™*, (Abu Dhabi, UAE), Dec 2018.

[9] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-Armed Bandit Learning in IoT Networks: Learning helps even in non-stationary settings," in *12th EAI Conference on Cognitive Radio Oriented Wireless Network and Communication*, CROWNCOM Proceedings, (Lisbon, Portugal), 2017.

[10] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[11] L. Besson and E. Kaufmann, "Multi-Player Bandits Revisited," in *Algorithmic Learning Theory*, (Lanzarote, Spain), Mehryar Mohri and Karthik Sridharan, 2018.

[12] E. Boursier and V. Perchet, "SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits," *arXiv preprint arXiv:1809.08151*, 2018.

[13] R. Kumar, S. J. Darak, A. Yadav, A. K. Sharma, and R. K. Tripathi, "Two-stage decision making policy for opportunistic spectrum access and validation on USRP testbed," *Wireless Networks*, pp. 1–15, 2016.

[14] R. Kumar, S. J. Darak, A. Yadav, A. K. Sharma, and R. K. Tripathi, "Channel Selection for Secondary Users in Decentralized Network of Unknown Size," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2186–2189, 2017.

[15] A. Maskooki, V. Toldov, L. Clavier, V. Loscrí, and N. Mitton, "Competition: Channel Exploration/Exploitation Based on a Thompson Sampling Approach in a Radio Cognitive Environment," in *EWSN-International Conference on Embedded Wireless Systems and Networks (dependability competition)*, (Graz, Austria), Feb 2016.

[16] V. Toldov, L. Clavier, V. Loscrí, and N. Mitton, "A Thompson Sampling Approach To Channel Exploration Exploitation Problem In Multihop Cognitive Radio Networks," in *PIMRC*, (Valencia, Spain), pp. 1–6, Sep 2016.

[17] L. G. Roberts, "Aloha packet system with and without slots and capture," *SIGCOMM Comput. Commun. Rev.*, vol. 5, pp. 28–42, Apr. 1975.

[18] J. R. Norris, *Markov Chains*, vol. 2 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998.

[19] X. Yang, A. Fapojuwo, and E. Egbogah, "Performance analysis and parameter optimization of random access backoff algorithm in lte," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sep. 2012.

[20] J.-Y. Audibert, R. Munos, and C. Szepesvári, "Tuning bandit algorithms in stochastic environments," in *International Conference on Algorithmic Learning Theory*, (Sendai, Japan), pp. 150–165, Springer, 2007.

[21] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *International Conference on Algorithmic Learning Theory*, pp. 174–188, Springer, 2011.

[22] Y. Cao, W. Zheng, B. Kveton, and Y. Xie, "Nearly Optimal Adaptive Procedure for Piecewise-Stationary Bandit: a Change-Point Detection Approach," in *AISTATS*, (Okinawa, Japan), 2019.

[23] S. J. Darak *et al.*, "Spectrum Utilization and Reconfiguration Cost Comparison of Various Decision Making Policies for Opportunistic Spectrum Access Using Real Radio Signals," in *CROWNCOM 2016*, (Grenoble, France), 2016.

[24] L. Besson, R. Bonnefoi, and C. Moy, "MALIN: an Implementation of Multi-Armed Bandits Learning Schemes for Internet-of-things Networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC) (IEEE WCNC 2019)*, (Marrakech, Morocco), April 2019. Following a Demonstration presented at International Conference on Telecommunications (ICT).