Upper-Confidence Bound for Frequency Selection in IoT Networks with Retransmissions

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

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

***Abstract*—Internet of Things (IoT) and in particular Low Power Wide Area (LPWA) technologies are being developed to enable long-range Machine-to-Machine (M2M) communications. It is considered a main driver for a vast variety of applications, where the need to fit a growing number of end-devices requires to design novel and more efficient access schemes. For instance, in smart grid communications, simple access schemes lead us to an increase of the latency, the number of collisions, and the power consumption. We consider devices communicate with the gateway using a wireless ALOHA-based protocol, and taking as inputs for the learning process, the number of successful transmitted packets and retransmissions. In this article, we propose and evaluate different learning strategies based on Multi-Arm Bandit (MAB) algorithms that allow the devices to improve their access to the network, while taking into account the impact of encountered collisions. Our heuristics try to retransmit in a different channel in case of collision in a channel first chosen by a UCB algorithm, a well-known MAB strategy. Empirical results show that approaches based UCB obtain a significant improvement in terms of successful transmission probabilities, even if the naive UCB is good and outperforms other strategies.**

***Index Terms*—ALOHA Networks, Internet of Things, Retransmissions, Multi-armed Bandits, Reinforcement Learning.**

I. Introduction

Nowadays, the Internet of Things (IoT) is considered a promising technology that will support the communications among a large number of devices, as it has become evident with the increasing demand of smart grids and smart cities projects. Nevertheless, the development of IoT networks also require the redesign of the entire paradigm of the legacy technology, since new research aspects such as the lower power consumption and low signaling need to be considered.

In this context, the LPWAN [1] technology has been conceived for providing the aforementioned features. For instance, LoRaWAN and SigFox technologies have been adopted in the monitoring of large scale systems (*e.g.*, smart grids), where a large number of devices compete for the transmission of their packets, in unlicensed ISM bands (*e.g.*, 433.5 MHz in Europe). Then, the improvement of the network performance in these unlicensed bands require to conceive novel MAC mechanisms.

One important concern in the MAC design is to reduce the Packet Loss Ratio (PLR) due to the interference caused by the collisions among the devices within the network and those following different standards. In fact, the number of collisions increases as more devices without coordination share the same band. Hence, novel access mechanisms considering the collisions need to be addressed to avoid degrading the network performance, while at the same time targeting features of IoT networks.

In this regard, Multi-Arm Bandit (MAB) algorithms [2] have been recently proposed as a solution and in particular in LPWA networks [3], [4], [5]. For instance in [5], the non-stationarity introduced by the presence of more than one intelligent object is addressed by MAB algorithms. In that work, low-cost algorithms following two well-known approaches, such as the Upper-Confidence Bound (UCB) [6], and the Thompson Sampling (TS) algorithms [7] have reported good results. Other recent directions include theoretical analysis of application of MAB algorithms for slotted wireless protocols in a decentralized manner, see [8] and references therein, but in this work we focus on a more programmatic approach.

The aim of this paper is to assess the performance of MAB algorithms [2], used for frequency selection in IoT networks. In particular, and compared to the literature, we focus on the impact of the retransmissions on the performance of learning algorithms. Indeed, in the case where learning algorithms are used for frequency selection, IoT devices tend to focus on a single channel, which increases the probability of having several successive collisions. The contributions of this paper can be summarized as follows:

- We first propose a closed form approximation for the probability of having a second collision after a collision has occurred in one channel,
- Then, we introduce several heuristics so as to cope with retransmissions,
- We finally conduct simulations in order to compare the performance of the proposed heuristics with the naive uniform random approach and the simple UCB strategy

This work is supported by the French National Research Agency (ANR), under the projects SOGREEN and EPHYL (grants coded: *N ANR-14-CE28-0025-02* and *N. ANR-16-CE25-0002-03*), by Région Bretagne, France, the French Ministry of Higher Education and Research (MENESR), and École Normale Supérieure de Paris-Saclay.

(non aware of retransmissions).

Finally, our proposal is applied in a decentralized manner, and with a low complexity that do not require any modification on network side, and it could be applied with a very low extra cost in real embedded hardware. Implementation of the model and our proposals on real hardware [9] is left as a future work.

The rest of the paper is organized as follows: first the system model is introduced in Section II. Section IV describes more formally the MAB learning algorithms. Our contributions mainly consist in heuristics, presented in Section V, while numerical results are presented in Section VI. Conclusions are given in Section VII.

II. System model

We suppose an IoT network made of one gateway (*i.e.*, base station), and of a large number of end-devices that regularly send short data packets to the gateway. The base station listens to a fixed set of K channels ($K > 1$), in which devices can transmit their packets. As in [3], we suppose that the network is made of two types of devices:

- We have a set of *static* end-devices. These devices are low-cost devices and are only able to use one channels. They use the same gateway as others. However, as we consider an IoT standard which operates in unlicensed bands, it could be considered that they communicate with another gateway or are using another standard, without changing much the model and our conclusion.
- We also have *dynamic* devices which have the possibility to use all the K available channels.

The IoT network considered here is a slotted ALOHA protocol [10], where each device has a probability $p > 0$ to transmit a packet in a slot (first transmission). In case of collision, a device retransmits this packet after a random waiting time, uniformly distributed in $[\![0; m-1]\!]$, where $m \in \mathbb{N}^*$ is the length of the back-off interval. We denote by $M_t \in \mathbb{N}^*$ the maximum number of transmissions for each packet. We assume that, in the case where more than two devices are transmitting a packet in the same slot, all the packets are lost and must be retransmitted.

With such assumptions, we can model the behavior of end-devices using the Markov chain [11] of Figure 1.

We consider $K > 1$ orthogonal radio channels (also called *arms*) of different characteristics, being unknown to the device. The radio protocol is slotted in both time and frequency, meaning that at each time step $t \in \mathbb{N}$, the device, *tries to* communicate in a channel $C(t) \in \{1, \dots, K\}$. In our model, the device chooses one channel k at a time, and use it to communicate, and waits for the gateway to send back an acknowledgement (*Ack*).

In the stochastic model considered in this paper, after choosing the arm k, receiving an acknowledgement provides a *reward* $r_k(t)$, randomly drawn from a certain distribution depending on the arm index. Rewards are assumed to be bounded in $[0, 1]$, and generally they follow one-parameter exponential families (a well-known example being any family of Gaussian distributions with a fixed variance). We present our algorithm by restricting to Bernoulli distributions[1] for sake of simplicity, meaning that arm k has a parameter $\mu_k \in [0, 1]$ and rewards are drawn from $B(\mu_k)$, $r_k(t) \sim B(\mu_k)$, which can be simply interpreted by the device: it is 1 if the channel k is not used by any other device during the time slot t, and is 0 otherwise.

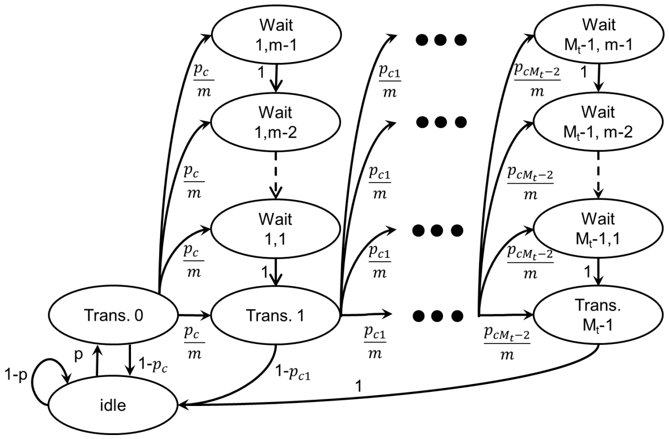

Fig. 1. The considered Markov model for the behavior of all the end-devices in the network. Transition are labeled by their probabilities, see [11].

III. Motivations for the proposed approach

We can justify why the proposed approach is interesting.

A. Analytic derivation

When MAB learning algorithms are used for the purpose of channel selection in an IoT network, the devices that implement them are learning to make most of their transmissions in only one channel. See for instance the numerical results presented in [5], or [12] for a different application example. However, in the case with many devices following the behavior described by Figure 1, if they are using the same channel, the probability p_{c1} to have a collision at the second transmission could be much higher than p_c, the probability of having a collision at the first transmission.

The aim of this section is to propose an approximation for p_{c1} as a function of p_c in order to assess the difference between these probabilities. To do so, we suppose that a device had a collision after transmitting a packet for the first time (transmission #0) and we compute the probability to have a collision at the second transmission (transmission #1).

Hypotheses We make two realistic hypotheses.

$(H1)$ The probability to have a collision at the second transmission, p_{c1}, can be decomposed into the probability of having a collision with a packet transmitted by devices involved in the collision at the first transmission, which

[1] The model is similar for other distributions, and we also tested our proposal with Gaussian distributions, with finite support in $[0, 1]$, and similar conclusions were observed. Non-discrete rewards $r_k(t)$ are interpreted as a relative communication efficiency, instead of binary available/busy information, but we do not cover this aspect in more details in this work.

is denoted p_{ca}, and the probability of having a collision with a packet transmitted by other devices, not involved in the previous collision. The number of devices involved in the previous collision is supposed to be small enough, compared to the number of devices in the channel, to consider that this second probability is equal to p_c.

$(H2)$ M_t is supposed to be large enough to consider that devices are hardly ever in this state. With this hypothesis, if a device is involved in the previous collision it retransmits its packet after a random back-off time.

We assume a steady state for the Markov chain of Figure 1 [11]. Let us consider one device in the channel. We denote x_t^i the probability that it is transmitting a packet for the $i+1$ time in a given slot (for $i \in \{0, \cdots, M_t - 1\}$), and $x_t = \sum_{i=0}^{M_t-1} x_t^i$ the probability that it is transmitting in a given slot. The probability that this device has a collision at the first transmission is denoted p_c and satisfies

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

Moreover, the probability $p_c(k)$ that it has a collision with k packets sent by k different devices ($k \geq 1$), at the first transmission, is equal to

$$p_c(k) = \binom{N-1}{k} x_t^k \, (1 - x_t)^{N-1-k} . \quad (2)$$

After a collision at the first transmission, the device retransmits its packet after a random back-off interval. Using hypothesis $(H1)$, the probability that it has a collision at the second transmission is

$$p_{c1} = p_{ca} + (1 - p_{ca})\, p_c. \quad (3)$$

So, we need to express p_{ca}, the probability to have a collision with a packet involved in the previous collision, as a function of p_c. Assuming hypothesis $(H2)$, p_{ca} is the probability that a device involved in the previous collision choose the same back-off interval. Thus p_{ca} is

$$p_{ca} = \sum_{k=1}^{N-1} p_{ca}(k), \quad (4)$$

where $p_{ca}(k)$ is the probability that, knowing that the device had a collision at transmission $\#0$, it had a collision with k packets, and that at least one of the k devices involved in the previous collision choose the same back-off interval. And thus

$$p_{ca} = \frac{1}{p_c} \sum_{k=1}^{N-1} \binom{N-1}{k} x_t^k \, (1 - x_t)^{N-1-k} \left[1 - \left(1 - \frac{1}{m}\right)^k \right] . \quad (5)$$

We now use $(H1)$ once again, assuming that the number of devices involved in the first collision is small compared to $N-1$, the first terms of the sum of equation (5) are predominant. Moreover, for these terms, k is small compared to $N-1$, so $N-1-k \approx N-1$. So,

$$\begin{aligned} p_{ca} &= 1 - \frac{1}{p_c} \sum_{k=1}^{N-1} \binom{N-1}{k} x_t^k \, (1 - x_t)^{N-1-k} \left(1 - \frac{1}{m}\right)^k , \\ &= 1 - \frac{(1 - x_t)^{N-1}}{p_c} \sum_{k=1}^{N-1} \binom{N-1}{k} x_t^k \left(1 - \frac{1}{m}\right)^k . \end{aligned} \quad (6)$$

We can use the binomial theorem to compute the sum in (6), and we obtain the expression of p_{ca}

$$p_{ca} = \frac{1}{p_c} - \left(\frac{1}{p_c} - 1\right) \left[1 + \left(1 - (1 - p_c)^{\frac{1}{N-1}}\right) \left(1 - \frac{1}{m}\right) \right]^{N-1} . \quad (7)$$

p_{c1} can finally be computed by inserting equation (7) in (3).

B. Numerical validation

In order to assess the proposed approximation, we suppose a channel in which all the devices have the same behavior. In this single channel, the ALOHA protocol is using a maximum number of transmissions of each message of $M_t = 10$, a back-off interval of maximum length $m = 10$, and a probability of transmission of $p = 10^{-3}$. Figure 2 shows the probability of collision in the channel, versus N the number of devices.

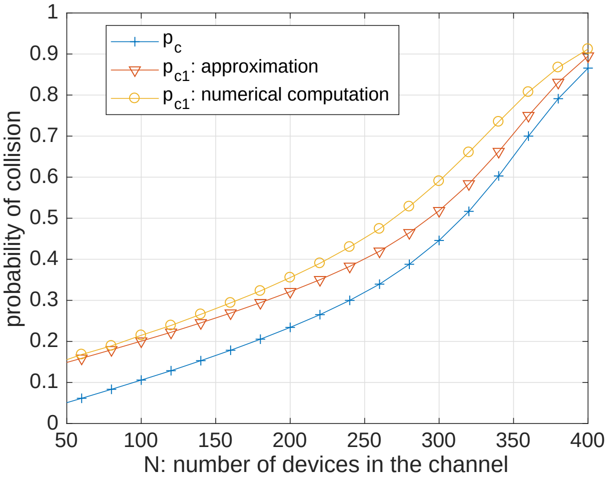

Fig. 2. Proposed approximation for the probability of collision at the second transmission. Our approximation is particularly precise for small values of N.

We can see in Figure 2 that the proposed approximation is precise where $p_{c1} \leq 30\%$, *i.e.*, where the gap between p_c and p_{c1} is the higher. Moreover, we can see in this figure that the gap between p_{c1} and p_c can be of up to 10%, which emphasizes the possible interest of improving MAB algorithms, so that they do not use the same channel for retransmissions.

IV. A well-known MAB Algorithm: UCB

Before presenting different ways to incorporate the retransmissions, we present a classical bandit algorithm as a base building block. We chose to restrict to the UCB algorithm [13], which is known to be efficient for stationary *i.i.d.* rewards while being simple to present, and being simple enough to be implemented in practice for embedded hardware. In this section, let us consider one device, which tries to learn how to reduce its PLR while accessing the network, without taking into account the retransmission aspect of our model. More details on this simpler variant can be found in previous work [5] and more theoretical details are given, *e.g.*, in [2], [8].

A. *Smart devices cannot be greedy*

A first approach is to 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 [14]. 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 succeed, the device will *never* use the first channel again, even it is the best one (*i.e.*, the most available, in average).

B. *The* UCB *algorithm*

Rather than relying on the empirical mean reward, Upper Confidence Bounds 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 channel k was selected up-to time $t \geq 1$,

$$N_k(t) = \sum_{\tau=1}^{t} \mathbb{1}(C(\tau) = k). \tag{8}$$

The empirical mean estimator $\widehat{\mu_k}(t)$ of channel k is defined as the mean reward obtained by selecting it up to time t,

$$\widehat{\mu_k}(t) = 1/N_k(t) \sum_{\tau=1}^{t} r_k(\tau)\mathbb{1}(C(\tau) = k). \tag{9}$$

For UCB, the *confidence* term is given by [13]

$$B_k(t) = \sqrt{\alpha \log(t)/N_k(t)}, \tag{10}$$

giving the upper confidence bound $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$, which is used by the device to decide the channel for communicating at time step $t+1$: $C(t+1) = \arg\max_{1\leq k\leq K} U_k(t)$. UCB is called an *index policy*.

The UCB algorithm uses a parameter $\alpha > 0$, originally α was set to $\alpha = 2$ [6], but empirically $\alpha = 1/2$ is known to work better (uniformly across problems), even though $\alpha > 1/2$ was advised by the theory [2]. In our model, every dynamic device implements its own UCB algorithm, *independently*. For one device, the time t is the total number of sent messages from the beginning, as rewards are only obtained after a transmission. Algorithm 1 presents the pseudo-code of UCB. Note that no matter if it is the first transmission or a retransmission of a message, this first proposal uses the same access scheme and updates the internal data in a similar way.

```
Input: Number of arms, $K \geq 1$
Input: Time horizon, $T \geq 1$, not used for the learning
Data: $N_k(t)$, $\widehat{\mu_k}(t)$, $B_k(t)$ and $U_k(t)$ for each $k$
Result: $C(t) \in \{1,\dots,K\}$ for each $t \in \{1,\dots,T\}$
for $t = 1,\dots,T$ do    // At every time step
    Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
    Transmit in channel $C(t) \sim \arg\max_k U_k(t)$;
    Reward $r_{C(t)}(t) = 1$ if Ack is received, else 0;
    Update internal data following Eq.(8), (9), (10);
end
```

Algorithm 1: A base building block, the UCB algorithm.

V. Proposed Heuristics

Machine learning algorithms, and especially the MAB framework, has been used in simpler cognitive radio models before, starting from [15] and more recently in [5], [3], for instance. The novelty of our approach relies on the proposed heuristics that try to take into account the retransmission aspect of our model.

In our model, any dynamic object knows when it is retransmitting or sending for the first time, but using the unmodified UCB algorithm for decision making (*i.e.*, channel selection) in both cases do not use this information. As usual in reinforcement learning, a natural question is to evaluate whether using this additional contextual information can improve the performance of the learning policy.

We present in this Section some heuristics, that use this information in various ways. They vary only in their way to select channels for retransmissions, and all retransmissions are dealt with similarly (no distinction is done between the first retransmission and the next $M_t - 1$ ones). All the following algorithms follow the same pattern as Algorithm 1.

A. UCB *then uniform random access*

This first proposal is a simple mixture between the UCB approach presented in Section IV-B and the naive Random Uniform Access approach, see Algorithm 2 below. It uses a UCB to select channels for the first transmission of each message, and in case of any retransmission, it uses a random channel selection. The idea is to learn to use the best channel for first transmission, then avoid using the best channel too much for the retransmissions. It is the simplest heuristic, inspired from the observations on p_{c1} being larger than p_c presented in Section III-B.

B. *Two* UCB

Another heuristic is presented in Algorithm 3: it tries to learn more, and it uses two different learning algorithms. As for all heuristics, one UCB algorithm is used to select the

```
for $t = 1, \dots, T$ do    // At every time step
    if First transmission of this message then
        Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U_k(t)$.
    else            // Random retransmission
        Transmit in channel $C(t) \sim \mathcal{U}(1, \dots, K)$.
    Reward $r_{C(t)}(t) = 1$ if Ack is received, else 0;
    Update internal data following Eq.(8), (9), (10);
end
```

Algorithm 2: UCB then Uniform Random Access.

channels for the first transmissions, but now the channels for retransmissions are not randomly selected but are selected using a second[2], independent, UCB algorithm. The idea here is that maybe the second algorithm will learn that retransmissions should happen in the other channels than the best one identified by the first algorithm. The second UCB uses data denoted, *e.g.*, $U'_k(t)$.

```
Data: $N'_k(t)$, $\widehat{\mu_k}'(t)$, $B'_k(t)$ and $U'_k(t)$ for each $k$
for $t = 1, \dots, T$ do    // At every time step
    if First transmission of this message then
        Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U_k(t)$.
    else  // Retransmission using 2nd UCB
        Compute $U'_k(t) = \widehat{\mu_k}'(t) + B'_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U'_k(t)$.
    Reward $r_{C(t)}(t) = 1$ if Ack is received, else 0;
    Update internal data following Eq.(8), (9), (10), for
      the first or second UCB algorithm;
end
```

Algorithm 3: Two UCB.

C. *One* UCB *then* K UCB

Extending the idea of the previous heuristics, we also propose a similar one, which uses $K + 1$ algorithms instead of simply two, see Algorithm 4. One UCB is again used for selecting channels for the first transmissions, and now K different and independent algorithms[3] are used for retransmissions, denoted $\mathcal{A}_1, \dots, \mathcal{A}_K$. If the first transmission happened in channel j, then the decision making for retransmitting is handled by algorithm $\mathcal{A}_j$. This algorithm $\mathcal{A}_j$ uses data denoted, *e.g.*, $U_k^j(t)$.

D. *Two* UCB *with delay for the second one*

The last heuristic we propose is a mixture of Algorithms 2 and 3, see Algorithm 5. Fix a delay Δ, *e.g.*, $\Delta = 100$ steps[4]. Then this last heuristic uses one UCB algorithm for first transmissions, and for the first Δ retransmissions, it selects channels uniformly at random. Only after Δ "naive" retransmissions, a second UCB algorithm is created and begins to be used for selecting channels for retransmissions.

```
Data: $\forall k, j \in [\![1; K]\!]$, $N_k^j(t)$, $\widehat{\mu_k}^j(t)$, $B_k^j(t)$ and $U_k^j(t)$
for $t = 1, \dots, T$ do    // At every time step
    if First transmission of this message then
        Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U_k(t)$.
    else  // Retr. after trying channel $j$
        Compute $U_k^j(t) = \widehat{\mu_k}^j(t) + B_k^j(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U_k^j(t)$.
    Reward $r_{C(t)}(t) = 1$ if Ack is received, else 0;
    Update internal data following Eq.(8), (9), (10);
end
```

Algorithm 4: UCB then K UCB.

```
Input: Delay $\Delta$, e.g., $\Delta = 100$
Data: $N'_k(t)$, $\widehat{\mu_k}'(t)$, $B'_k(t)$ and $U'_k(t)$ for each $k$
for $t = 1, \dots, T$ do    // At every time step
    if First transmission of this message then
        Compute $U_k(t) = \widehat{\mu_k}(t) + B_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U_k(t)$.
    else if Less than $\Delta$ retransmission then
     // Random retransmission
        Transmit in channel $C(t) \sim \mathcal{U}(1, \dots, K)$.
    else  // Retransmission using 2nd UCB
        Compute $U'_k(t) = \widehat{\mu_k}'(t) + B'_k(t)$;
        Transmit in channel $C(t) \sim \arg\max_k U'_k(t)$.
    Reward $r_{C(t)}(t) = 1$ if Ack is received, else 0;
    Update internal data following Eq.(8), (9), (10);
end
```

Algorithm 5: UCB then Uniform Random Access.

VI. Experiment Results

We simulate our network in order to compare the proposed heuristics. The simulated network uses similar values of the ALOHA model as the one used for Figure 2: $\forall t, M_t = 5$, $m = 10$, and $p = 10^{-3}$. We consider $K = 4$ channels (like for the LoRa standard), a number of time slots 1000000, which is large enough to observe convergence of all learning algorithms. Moreover the results are averaged on 10 independent runs of the random simulation. We consider a total number of devices of $N = 2000$, and a non-uniform repartition of static devices of $40\%, 30\%, 20\%, 10\%$ in the 4 channels.

We present in Figure 3 the results of this numerical simulation. The x axis corresponds to the number of communications

[2] The storage requirements and time complexity is doubled but remains linear w.r.t. K, and so it is still a practical proposal.

[3] The storage requirements and time complexity is now quadratic in K, and as such we no longer consider this heuristic to be a practical proposal in some IoT networks, as for instance Sigfox networks consist in a large number of very narrow-band channels. But for LoRaWAN networks with $K = 4$, storing $K + 1 = 5$ algorithms does not cost much more than storing 2.

[4] Choosing the value of Δ could be done by extensive benchmarks but such approach goes against the reinforcement learning idea: an heuristic should work against any problem, without the need to be able to simulate the problem in advance in order to find a good value of some internal parameter. As such, we only consider a delay of $\Delta = 100$.

of a device, and it corresponds to the number of learning step, while the y axis corresponds to the successful transmission rate, in percentage.

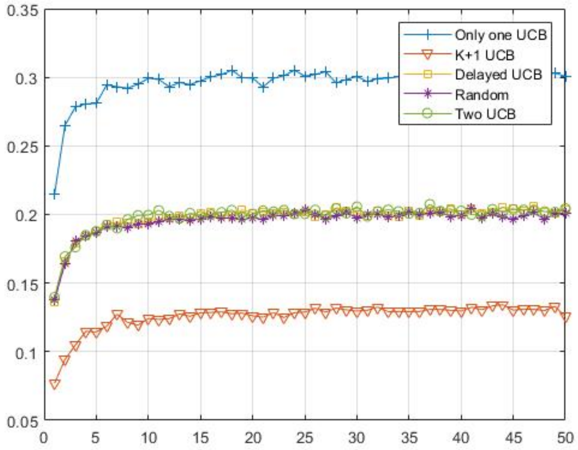

Fig. 3. Comparison of different heuristics based on UCB, and the vanilla UCB algorithm. "Random" refers to UCB then random retransmission.

We verify that each proposal is indeed learning, as its successful transmission rate is rapidly increasing (or equivalently, its PLR is decreasing). All plots show a pattern typical of MAB algorithms when applied to this kind of problem: a fast learning phase in the beginning of the experiment and a "plateau" showing the convergence of the algorithm to a stable strategy.

All heuristics outperform the naive random uniform approach, which is not included to reduce clutter in the plot. The random approach has a successful transmission rate of about 7%, constant in time as no learning is involved, and for instance the best heuristic attains a performance up-to 4 times better, at about 30%. Observing such a strong improvement in terms of successful transmission rate is a very strong advocate of using simple MAB learning algorithm.

The conclusions we can draw from this simulation are twofold. First of all, a simple sanity check is that all the proposed heuristics do not reduce performance when compared to the naive approach. But most importantly, we observe three groups of heuristics. The less efficient one is the $K+1$ UCB, and this makes sense as having more learning algorithm needs more for each of them to learn. The more efficient one is the simple UCB procedure, and this was quite surprising. All the other heuristics perform very much similarly.

VII. Conclusion

Summary

In this paper, we presented a model of IoT networks based on a ALOHA protocol, slotted both in time and frequency, in which dynamic objects can use machine learning algorithms to improve their Packet Loss Ratio when accessing the network. The main novelty of this model is that it allows device to retransmit a packet in case of collision, and by using the framework of Multi-Armed Bandit, we presented and evaluated several learning heuristics that try to learn how to transmit and retransmit in a smarter way. Empirical simulations show that each heuristics outperform the naive uniform access scheme, and we conclude that the simple UCB learning approach is the most efficient.

Future works

Possible extensions include studying other families of algorithms, such at Thompson Sampling [7], or non-stochastic MAB algorithms such as the Exp3 family [2]. We also want to study more heuristics, and one interesting idea is to consider variants of stochastic MAB algorithms such as Sliding Window UCB or Discounted UCB [16], or smarter and more recent algorithms. As they are tailored to be robust to non-stationary MAB problems with small dynamic behaviors, applying our heuristics to such base algorithm could help to be more adaptive to the intrinsic non-stationarity of the network.

A last direction of future work includes a real-world hardware implementation of this model, in order to validate experimentally our results. We have the experience to do it very soon, using Ettus USRP boards [9], the GNU Radio and GNU Radio Companion software [17], and our TestBed[5].

References

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 855–873, Secondquarter 2017.

[2] 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.

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

[4] A. Azari and C. Cavdar, "Self-organized Low-power IoT Networks: A Distributed Learning Approach," in *IEEE Globecom 2018*, 12 2018.

[5] 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, 2017.

[6] 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.

[7] 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.

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

[9] "USRP Hardware Driver and USRP Manual." http://files.ettus.com/manual/page_usrp2.html. Accessed: 2018-09-25.

[10] N. Abramson, "The ALOHA System: Another Alternative for Computer Communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), (New York, NY, USA), pp. 281–285, ACM, 1970.

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

[5] Cf. the page `www-scee.rennes.supelec.fr/wp/testbed/` and the associated publications, it is also presented in details in [18].

[12] V. Toldov *et al.*, “A Thompson Sampling approach to channel exploration-exploitation problem in multihop cognitive radio networks,” in *PIMRC*, pp. 1–6, 2016.

[13] 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.

[14] T. L. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[15] W. Jouini, D. Ernst, C. Moy, and J. Palicot, “Upper Confidence Bound based decision making strategies and Dynamic Spectrum Access,” in *IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, 2010.

[16] A. Garivier and E. Moulines, “On Upper-Confidence Bound Policies For Non Stationary Bandit Problems,” *arXiv preprint arXiv:0805.3415*, 2008.

[17] “GNU Radio Documentation.” https://www.gnuradio.org/about/. Accessed: 2018-09-25.

[18] Q. Bodinier, *Coexistence of Communication Systems Based on Enhanced Multi-Carrier Waveforms with Legacy OFDM Networks*. PhD thesis, CentraleSupélec, 2017.

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`.