Computer Communications 196 (2022) 129–140

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

LoRa network reconfiguration with Markov Decision Process and Fuzzy C-Means clustering

Aghiles Djoudi [a,b,*], Rafik Zitouni [b,c], Nawel Zangar [a], Laurent George [a]

[a] LIGM, University Gustave Eiffel, CNRS, ESIEE Paris, Marne-la-Vallée, France
[b] ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France
[c] 5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey, UK

ARTICLE INFO

Keywords:
Fuzzy C-Means
Markov Decision Process
Q-learning
LoRaWAN

ABSTRACT

Long Range (LoRa) is a proprietary modulation technique that uses Chirp Spread Spectrum (CSS) modulation for low power and wide area communications. Despite the advantages of LoRa technology, the reconfiguration of transmission parameters such as Spreading Factor (SF) and Transmission Power (P^{tx}), remains limited to maximize the uplink traffic. In this paper, we look upon additional parameters such as the Bandwidth (BW) and the Coding Rate (CR). We apply Fuzzy C-Means (FCM) algorithm to acquire knowledge about the quality of each transmission setting. Then, we use this knowledge in Q-learning and Markov Decision Process (MDP) algorithms as a state transition matrix to converge better and faster to the set of transmission settings that maximize the uplink data rate. As the solution should cope with different scenarios, we vary the number of End Devices (EDs), Base Stations (BSs), Packet Sizes (PSs) and Packet Rates (PRs). In addition, we compare our solution with many algorithms such as EXP3, ADR and EXPLoRaTS. Simulation results show that MDP with FCM clustering preprocessing improves better several Quality of Service (QoS) metrics including the Data Rate (DR), Packet Delivery Ratio (PDR), Time on Air (ToA) and Transmission Energy (E^{tx}). Thus, the PDR and the DR were improved by 25%, the ToA was reduced by 40% and E^{tx} was reduced by 20%.

1. Introduction

Unlicensed bands are more and more used by all kinds of wireless technologies (Wi-Fi, LTE-U, ZigBee, Z-Wave, X = tooth, LoRaWAN, Sigfox, Ingenu, Weightless, etc.). This heavy use of unlicensed bands will certainly cause performance decay due to contention problems. Efficient Channel Access Control (MAC) protocols allow devices to avoid such behavior by exchanging extra control messages (signaling overhead). However, due to the high energy consumption required to run such protocols in Internet of things (IoT) devices, new approaches should be investigated using simple ALOHA-based mechanisms [1] with a lower signaling overhead. In this article, we adjust LoRa transmission settings to enhance the quality of uplink traffic using both Q-learning and MDP with FCM clustering pre-processing.

We aim in this work to maximize the uplink traffic of LoRa devices by comparing it to solutions proposed in the literature such as Long Range Wireless Access Network (LoRaWAN) alliance that proposed Adaptive Data Rate (ADR) algorithm [2]. Their mechanism adjusts periodically the and the SF according to the Received Signal Strength Indication (RSSI). However, this algorithm [2] shows a lack of mechanisms to maximize uplink data transmissions. In fact, to be able to maximize well the DR, the uplink state between each end-device and the Gateway (GW) should be measured and characterized in advance to speed up the convergence and achieve better DR. For this reason, we characterize the link state of each end-device using FCM algorithm [3,4] based on the transmission settings used to send each packet.

To deal with the randomness of the wireless environment, the research community has a consensus that future networks must be flexibly designed to deal with this challenge. Therefore, we use, in this work , an online reconfiguration of transmission settings to make the network smart enough to converge by itself to the set of parameters that best fit environment conditions. For this reason, Reinforcement Learning (RL) algorithms are good candidates to reinforce the selection of the suitable transmission settings at each transmission. We propose in this work to enhance the uplink data rate by clustering transmission settings to know at which quality level each setting could lead. Then we feed this knowledge, in the form of membership degrees, to MDP as a state transmission matrix. Thus, transmission settings will lead to different states based on their membership degrees to different clusters. This means that when we pickup one setting from a cluster, FCM is able to recognize at which link quality level the uplink traffic will be expected to jump if we select such an action.

* Corresponding author at: LIGM, University Gustave Eiffel, CNRS, ESIEE Paris, Marne-la-Vallée, France.
E-mail addresses: aghiles.djoudi@esiee.fr (A. Djoudi), nawel.zangar@esiee.fr (R. Zitouni), laurent.george@esiee.fr (N. Zangar), r.zitouni@surrey.ac.uk (L. George).

https://doi.org/10.1016/j.comcom.2022.08.020
Received 26 January 2022; Received in revised form 12 July 2022; Accepted 23 August 2022
Available online 6 September 2022
0140-3664/Published by Elsevier B.V.

In this paper, we investigate the problem of LoRa transceivers' reconfiguration to enhance the quality of the uplink traffic. Our main contributions are as follows:

- We characterize each transmission setting by measuring their membership degrees to different QoS levels based on the FCM algorithm.
- We use FCM membership degrees in MDP algorithm as state transition probabilities between uplink states.
- Through intensive simulations, we assess and we compare the DR, the PDR, the ToA and the of both Q-learning and MDP with Exponential weights for Exploration and Exploitation (EXP3), ADR and EXPLoRaTS algorithms.

The key contributions of this paper are further reported as follows. First, Section 2 and Section 3 elucidate summary of related works and the required background about LoRa transmission settings that we aim to optimize. Section 4 enunciates the problem statement. In Section 5 and Section 6, we explain how MDP with FCM are applied to select the suitable set of transmission settings. Simulation settings are presented in Section 7 and our findings are highlighted in Section 8. Finally, Section 9 concludes this paper.

2. Related work

To overcome the limitation of the default ADR scheme of LoRaWAN alliance [2] that suffers from a weak DR, many works in literature propose to use either heuristic or machine learning algorithms. For example, two different SF allocations algorithms (EXPLoRaSF and EXPLoRaTS) are presented in [5] as an alternative to ADR. To allow better equalization of the ToA among the SF channels, the proposed algorithms select the best SF based on the number of connected devices, the distance and the RSSI. Specifically, authors attempt to use a high DR to offload the traffic of the less congested highest SF. EXPLoRaSF aims to efficiently distribute the SF among end-devices. It selects the SF according to the total number of connected devices. Particularly, it equally allocates SFs to n devices based only on the RSSI, where the first $n/6$ devices with the highest RSSI get the SF 7 and then the next $n/6$ devices get the SF 8, etc. EXPLoRaTS is more dynamic than EXPLoRaSF since it equalizes the ToA of packets that are transmitted with different SFs.

Since LoRaWAN is a Pure ALOHA-based network, mutual interferences and traffic collisions caused by the random access to the channel decrease Significantly the network throughput. To deal with such a problem, Aihara et al. [6] propose a wireless resource allocation scheme with Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol. They try to maximize the weighted sum of PDRs using Q-learning. Their results show that each device can avoid packet collisions without an explicit feedback. The average PDR is then improved by about 20% compared to the random allocation scheme. However, the obtained results are compared only with random transmissions even if other algorithms exist.

In the same context, a deep reinforcement learning (DRL) has been applied for LoRaWAN transmission settings by Ilahi et al. [7]. Their algorithm mitigates collisions and outperforms the state of the art learning-based techniques achieving up to 400% improvement of PDR. However, this work need to be extended to prove the efficiency of DRL-based solutions in more scenarios and not only with one gateway and less than 100 devices. In addition, as the authors manage to assign transmission parameters, other transmission parameters such as the BW, the CR and the need to be considered in the learning process.

Always in the context of machine learning approaches, a multi-agent Q-learning algorithm has been applied by Yu et al. [8] to allocate jointly the SFs and the to increase the uplink DR. Their algorithm is based on the interactions between the agent and the wireless environment. The agent updates dynamically its policy to optimize the reliability and the energy consumption. The authors have compared their algorithm with a static allocation mechanism. Simulation results show the advantages of using Q-learning algorithm to enhance the Signal-to-interference & noise ratio (SINR), the DR and the . However, this work missed other transmission parameters like the CR and the BW that need to be considered to exploit the efficiency of machine learning algorithms. In addition, only one scenario has been studied with a fixed number of devices, gateways and fixed packets' size and rates of transmissions.

Another solution to increase LoRaWAN uplink traffic has also been proposed by Ta et al. [9] but in decentralized management way. This means that the learning process is made by the end devices and not by the network server. The authors propose to use Multi-Armed Bandit (MAB) algorithm: EXP3, for the learning process. They try to increase the PDR while keeping the energy consumption as low as possible. However, such an approach make IoT devices consume a lot of energy for the training process which is not suitable for Low Power Wide Area Networkss (LPWANs)

Marini et al. [10] present a new MATLAB-based simulator tool covering the physical and the upper layers of LoRaWAN stack. The proposed simulator supports multiple gateways with receiving window prioritization. They investigated the impacts of different coding rates on interference probabilities. However, the advantage of increasing the coding rates comes at the expense of an increased energy consumption. They showed that the network operates in heavy interference conditions when the number of end devices increases. Increasing the CR in such conditions is then counterproductive. They assessed also the impact of downlink transmissions on the average energy consumption of devices. They showed that increasing the number of gateways affects not only the packet delivery rates but also the average energy consumption. However, they limited their work on building a simulation tool without any enhancement of the technology.

A Hybrid Adaptive Data Rate (HADR) control has been proposed recently by Farhad and Pyun [11] to increase the uplink traffic when there are both fixed and moving devices. Their approach helps to increase packet success rates by selecting the best MAC protocol according to applications constraints. For example, with environment monitoring applications, devices need to synchronize their transmission by using Time Division Multiple Access (TDMA) based access control scheme to avoid collisions. However, with event monitoring application, devices need to notify immediately with a random-access protocol. Thus, TDMA is not well suited for the second application and the algorithm should select the random-access protocol. However, similar contributions have already been proposed for WiFi and cellular network. In addition, signaling overhead caused by such an approach to synchronize the selection of the best protocol could drastically collapse the network especially in a dense network.

A new machine learning based scheme has been proposed by Azizi et al. [12] to increase the PDR. They propose to use RL based resource allocation algorithm enabling LoRa devices to adjust their transmission settings in a distributed manner. They proposed a MIX-MAB algorithm which combines two MAB schemes: Successive Elimination (SE) and EXP3 Ta et al. [9]. They highlight the advantage of their solution by comparing it with EXP3 algorithm in five different scenarios. Their simulation results show that they reduce the convergence time to the half of what EXP3 do while achieving a higher PDR. However, such a comparison has been made using 5 scenarios with the same number of devices, gateways and the same packet sizes. In addition, authors compared only the PDR missing the most important metric which is the data rate. Furthermore, computational overhead caused by their distributed training will not only reduce life time duration of the network but also increase channel occupancy of the Industrial, Scientific and Medical (ISM) band.

All previous works including [13,14] proposed in the literature to maximize the data rate prove their weakness since they did not measure the impact of each transmission setting on the uplink state before starting their learning process. They focus only on a limited number

of parameters without an explicit assessment of transmission settings qualities in advance. Thus, they start their learning process without any knowledge of the expected quality each transmission setting which leads to a high exploration rate rather than a high exploitation of the best transmission settings. We propose in this work to first recognize transmission settings qualities through a pattern recognition process and then we exploit directly the best transmission settings using MDP and the recognized patterns. In other words, the patterns recognized by FCM clustering process are given as input to Q-learning and MDP algorithms through a state transition matrix to know at which state each action is expected to lead. Furthermore, we propose to cover many scenarios with different numbers of devices and gateways and different packet sizes and rates. Then, we compare our findings with the previous presented algorithms: ADR, EXPLoRaTS and EXP3.

3. Background

LoRa is a proprietary modulation scheme derived from Chirp Sequence Spread Spectrum (CSSS) modulation whose main objective is to improve the RSSI at the expense of the Data Rate. It uses orthogonal SFs and allows to find a trade-off between the DR and the coverage. It is a physical layer implementation and does not depend on higher layer protocols. This allows it to coexist with different network architectures.

3.1. Transmission parameters

As mentioned above, LoRa devices, can be configured to use different , SFs, BWs and CRs to achieve the best connection performance with the highest data rate. The combination of these parameters results in around 6720 possible settings [15], allowing devices to fully adjust LoRa modulation to IoT application requirements. A brief description of each of the mentioned parameters is given below.

- **:** It can range from −2 dBm to 20 dBm, but due to implementation limits, it can be adjusted only from 2 dBm to 14 dBm for industrial products. To reduce radio pollution, a duty cycle less than 1% is required by LoRaWAN alliance.
- **Carrier Frequency (CF):** It is the central frequency that can range from 137 MHz to 1020 MHz, depending on the region of use.
- **Spreading Factor (SF):** It can range from 7 to 12 and is presented as the ratio between Symbol Rate (SR) and Chip Rate (HR): SF = log_2(HR/SR). A higher SF not only enhances the Signal to Noise Ratio (SNR), the range and the receiver sensitivity, but also the ToA. Each increase in the SF divides the transmission rate by two and doubles the transmission time and energy consumption.
- **Bandwidth (BW):** It is the range of frequencies in the transmission band. A higher BW offers a higher data rate (less airtime), but a lower resiliency to noise. Transmitted packets are sent with a chip rate equal to the BW. For example, the BW 125 kHz is equivalent to a chip rate of 125 kcps. A typical LoRaWAN network operates on: 125 kHz, 250 kHz, or 500 kHz.
- **Coding Rate (CR):** It is the Forward error correction (FEC) used by LoRa devices against interference and can be configured with: 4/5, 4/6, 4/7 and 4/8. A higher CR offers more protection against noise, but increases the ToA. Transmitters with different CRs can communicate since the CR of packets' header is always 4/8 encoded.

3.2. QoS metrics

To enhance the experience of IoT network users, application's requirements such as the DR, the PDR and the ToA should be improved to allow end-devices to transmit their data to the cloud in good conditions. Thus, future network optimization schemes should deal with the diversification of IoT applications. We report in this subsection the related QoS metrics used in our work.

- **Data Rate (DR):** It is the relationship between the desired Data Rate (DR), Symbol Rate (SR), Coding Rate (CR), and the desired Chip Rate (HR) (or bandwidth). It is expressed as follows:

$$\begin{aligned} \text{SR} &= \frac{BW}{2^{SF}} \\ \text{BR} &= SR \cdot SF \\ \text{DR} &= BR \cdot CR \end{aligned} \tag{1}$$

- **Time on Air (ToA):** It measures the transmission delay taken by one packet to reach the Gateway [16]. The ToA is computed using the Eq. (2) given by [17]:

$$ToA_{[s]} = \frac{2^{SF}}{BW}\left((NP + 4.25) + (SW + \max(J, 0))\right) \tag{2}$$

with:

$$J = \left\lceil \frac{8PS - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)} \right\rceil (CR + 4)$$

where:

  - NP = 8 if LoRa, 5 if GFSK
  - SW = 8 if LoRa, 3 if GFSK
  - CRC = 1 if uplink packet, 0 else
  - IH = 0 if header, 1 else
  - DE = 1 if ADR active, 0 else

- **Energy consumption:** It measures the energy consumed to transmit each packet. It is computed using the following equation:

$$E^{tx}_{[j]} = ToA_{[s]} * P^{tx}_{[w]} * 3.0_{[v]} \tag{3}$$

4. Problem statement

We formulate the online selection of the suitable set of configurations as an exploration/exploitation dilemma. We propose to maximize the utility of the network to enhance the quality of uplink traffic. As the main goal of LoRa end-devices is to send their collected data to the cloud with the highest data rate, network utilization (or utility) function is expressed as the Data Rate of the up-link traffic after each transmission at time t given by Eq. (4).

$$U_t = \begin{cases} DR_t & \text{if packet received} \\ 0 & \text{else.} \end{cases} \tag{4}$$

While fulfilling the utility requirements, this strategy will maximize the utilization of the scarce radio resources. However, finding the transmission settings that maximize the utility function is an NP-hard problem [18] for a practical size of network. Thus, to have a lower complexity, we use RL algorithms to converge analytically to the optimal transmission settings that maximize the network data rate. Proceeding this way, the Network Server (NS) will be able to learn which transmission setting fits well environment conditions and devices location and updates its configuration accordingly.

With RL algorithms, an agent tries to obtain as much reward as possible by carrying out the most rewarding action among N actions. For example, in MAB algorithms [19], the rewards of actions are randomly generated according to an unknown distribution. Therefore, they try to minimize the regret values (due to exploration of new actions) to find the most rewarding arms.

We focus in this work on applying both Q-learning algorithm and policy iteration of MDP algorithm to measure the quality of actions via Q-values. With Q-learning, an agent updates the quality of actions (Q-values) and learns the best policy by exploiting the previous actions and exploring new ones. In the other side, MDP updates its Q-values based on an initial knowledge of the environment provided by the state transition matrix. To initialize this knowledge, the membership matrix of FCM algorithm is transformed to a state transition matrix in Q-learning and MDP algorithms to disclose at which state each action will probably lead (see Fig. 1).

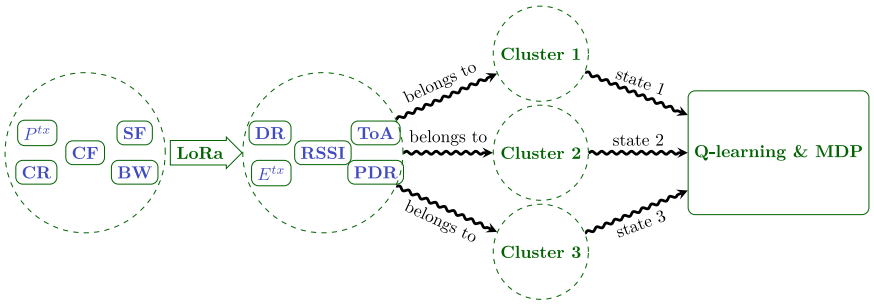


Fig. 1. LoRaWAN reconfiguration scheme.

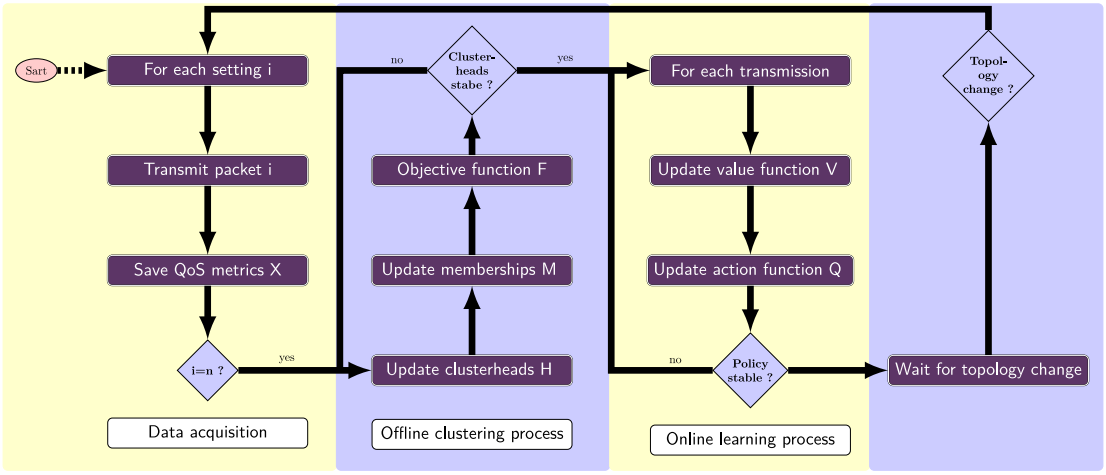


Fig. 2. Overview of functions.

An overall diagram of our three-step learning process is described in Fig. 2. First, we start by acquiring enough data to assess the quality of each transmission setting by sending randomly several packets with different transmission settings. Next, we apply the clustering process on the measured QoS metrics to extract patterns related to the quality of each transmission setting in offline mode. And then, we use this patterns as a state transition matrix in MDP to know at which state each transmission setting could lead. Section 5.5 gives more details about the state transition matrix and how MDP updates its functions to converge analytically to the optimal settings in online mode.

5. Markov Decision Process and Q-learning algorithms based on FCM clustering

To illustrate the learning process, Fig. 3 shows LoRaWAN architecture and the interactions between EDs, the NS through one or more GWs. The link between EDs and the GWs represents the uplink transmissions with a given transmission setting a. In the downlink side, the NS sends a new suggestion of parameters a' to enhance the previous uplink DR. With MDP, the probability to jump from one state to another is known from FCM through its membership degrees matrix. Whereas, as Q-learning requires to recognize at which new state each action will probably lead after each transmission to update its Q-values, we return the number of the cluster at which the selected transmission setting belongs more (see Fig. 4).

Interactions between the NS and the wireless environment are formally defined as a finite Markov Decision Process (MDP). We note: S a set of states that match the recognized patterns (Quality levels) by the clustering process. A a set of actions that match the transmission settings that we want to optimize. P a state transition function: $S \times A \times S \rightarrow [0-1]$, where $P(s,a,s')$ gives the probability to jump from state s to s' by taking action a, and finally a reward function R: $S \times A$,

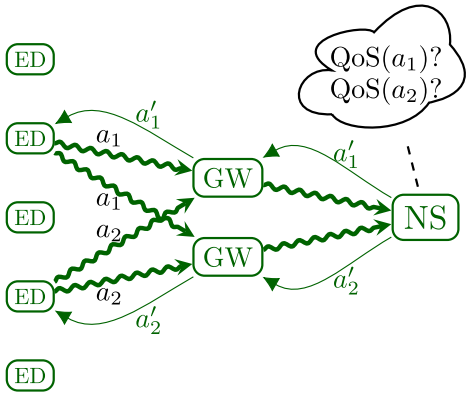


Fig. 3. LoRa WAN architecture.

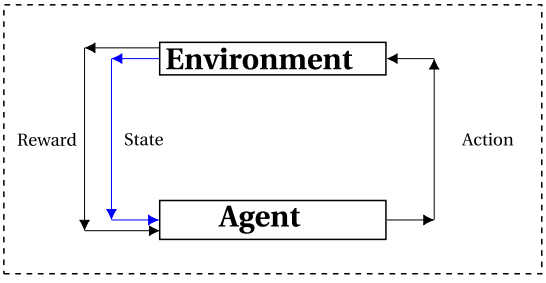


Fig. 4. Reinforcement learning process.

Table 1
Notations for MDP.

Notations for Q-learning and Markov Decision Process	
A	Set of actions
a	An action
R	Set of all possible rewards, a finite subset of $\Re$
R_t	The reward at time t
t	Discrete time step or play number
S	Set of all states
$A(s)$	Set of all actions available in state s
s, s'	States
π	Policy (decision-making rule)
$\pi(s)$	Action taken in state s with policy π
$\pi(a\|s)$	Probability of taking action a in state s with policy π

where $R(s,a)$ gives the NS a reinforcement feedback for the state–action pair (s,a).

- S=$\{s_0,\ldots,s_n\}$ is a finite set of states which in our study is a set of uplink state levels.
- A=$\{a_0,\ldots,a_n\}$ is a finite set of actions which in our study is a set of possible transmission settings.
- P=$(s_{t+1}=s'|s_t=s,a_t=a)$ is the transition probability from state s at step t to state s' at the next step due to an action a.
- R(s,a) is the expected reward received by applying an action a. It represents the gain of DR by improving our utility function.
- $\gamma \in [0,1]$ is called a discount factor, it represents the extent at which old rewards should be considered.

Table 1 summarizes the notations used in both Q-learning and MDP algorithms. The following subsections report how we calculate the cumulative reward and how we define both state and action value functions and the state transition matrix.

5.1. Cumulative discounted reward of MDP

The cumulative long-term discounted reward of state s at time t is the discounted sum of rewards that could be earned in this state and is given by $G_t(s)$. Each action yields a reward for the current state and represents the earned rewards during the learning process, denoted $R_t(s,a)$. The cumulative rewards earned when taking action a at state s is given by $G_{t+1}(s,a)$.

$$\begin{aligned} G_t(s) &\doteq \sum_{a'\in A(s)} G_t(s,a') \\ G_{t+1}(s,a) &\doteq \sum_{s'\in S_{t+1}} R_{t+1}(s,a) + \gamma \cdot G_t(s') \end{aligned} \tag{5}$$

We define the reward function as the difference between utility functions.

$$R_{t+1} = U_{t+1} - U_t \tag{6}$$

Where γ is the discount factor $(0<\gamma<1)$, which determines the impact of old rewards on the learning process. If $\gamma=0$, the agent will be "myopic" and will be focused on maximizing immediate rewards only, while forgetting all previous observations of rewards. In our case and after several experimental studies, we set γ to 0.9 as its the value that offers the highest learning rate considering 90% of the previous rewards observations.

The reward function R_{t+1} measures the data rate gained at each iteration. After performing an action a', the reward function measures how much data rate each device won compared to the previous action a. Both Q-learning and MDP aim to learn the policy (*i.e.* the sequence of actions) that earns more rewards, *i.e.* more DR, until there will be no possible data rate to win.

Algorithm 1: Policy iteration algorithm of MDP based on FCM.

```
Sates S = {1, ..., n_x}   ▷ Set of recognized patterns (Uplink quality levels)
Actions A = {1, ..., n_a},   A : S ⇒ A   ▷ Set of transmission settings
Reward function R : S × A → ℝ   ▷ The gain of data rate observed
Transition function P : S × A → S   ▷ Membership degrees of FCM
procedure PolicyEvaluation(S, A, R, P)   ▷ Policy Evaluation
    Δ ← ∞
    while Δ > ϵ do   ▷ While V is not stable
        for each s ∈ S do   ▷ For each state/cluster
            V_{t+1}(s) ← Σ_{s',r'} P(s', r | s, π(s)) [r + γV(s')]   ▷ Ubdate V
            Δ ← max(Δ, |V_t(s) − V_{t+1}(s)|)
        end for
    end while
end procedure
procedure PolicyImprovement(S, A, R, P)   ▷ Policy Improvement
    for each s ∈ S do   ▷ For each state
        π_{t+1}(s) ← arg max_a Σ_{s',r'} P(s', r | s, a) [r + γV(s')]   ▷ Eq. (10)
        if π_t(s) == π_{t+1}(s) then
            return V* ≈ V and π* ≈ π
        else
            PolicyEvaluation(S, A, R, P)
        end if
    end for
end procedure
```

5.2. State-value function of Markov Decision Process (MDP)

The state-value function [20] of an arbitrary policy π is expressed in the following equation.

$$\begin{aligned} V_t^{\pi}(s) &\doteq \mathbb{E}^{\pi}\left[G_t(s) \mid s_t = s\right] \\ &= \mathbb{E}^{\pi}\left[\sum_{a'\in A(s)} G_t(s,a') \mid s_t = s\right] \\ &= \mathbb{E}\left[\sum_{a'\in A(s)} Q_t^{\pi}(s,a') \mid s_t = s\right] \\ &= \sum_{a'\in A(s)} \pi(a' \mid s)\cdot Q_t^{\pi}(s,a') \end{aligned} \tag{7}$$

Where $\pi(a|s)$ is the probability to select action a at a state s. $Q(s,a)$ denotes the estimated cumulative reward earned when action a is selected at state s. The state value function of state s is the estimated reward that could be earned by taking an action in this state. As $E(x)=P(x)\cdot x$, the state value function denotes also the sum of rewards that could be earned by taking action $a\in A(s)$ weighted by its probability to occur.

The main objective of the learning process is to find the optimal policy $\pi^{\star}$, which is a mapping from S to A that maximizes the expected long-term discounted reward for each state. We note $\prod^{\pi}(s)=s_1,s_2,\ldots,$ the trajectory of the learning process when strategy π is used.

5.3. Action-value function of Markov Decision Process (MDP)

MDP has the ability to provide the network with necessary cognitive capabilities to build a transmission setting strategy according to environment conditions. The action value function [20] of action a represents the estimation of cumulative rewards that will be observed by taking this action. These rewards are expressed as the estimation of the sum of the rewards of the current action a and the estimated

rewards in the next state s_{t+1} (value function of state s_{t+1}). To get this estimation, this sum is weighted by the probability to jump from state s to $s' \in S_{t+1}$ using action a.

$$\begin{aligned} Q^{\pi}_{t+1}(s,a) &\doteq \mathbb{E}^{\pi}\left[G_{t+1}(s,a) \mid s_t = s, a_{t+1} = a\right] \\ &= \mathbb{E}^{\pi}\left[\sum_{s' \in S_{t+1}} r + \gamma \cdot G_t(s') \mid s_t = s, a_{t+1} = a\right] \\ &= \mathbb{E}\left[\sum_{s' \in S_{t+1}} r + \gamma \cdot V^{\pi}_t(s') \mid s_t = s, a_{t+1} = a\right] \\ &= \sum_{s' \in S_{t+1}} P\left(s' \mid s,a\right) \cdot \left[r + \gamma \cdot V^{\pi}_t\left(s'\right)\right] \end{aligned} \tag{8}$$

$$with\ r = R_{t+1}(s,a)$$

$$\begin{aligned} Q^{\star}_{t+1}(s,a) &= \sum_{s' \in S_{t+1}} P\left(s' \mid s,a\right) \cdot \left[R_{t+1}(s,a) + \gamma \cdot V^{\star}_t\left(s'\right)\right] \\ V^{\star}_t(s) &= \max_{a'} Q^{\star}_t(s,a') \end{aligned} \tag{9}$$

$$\pi^*(s) = \arg\max_{a \in A(s)} Q(s,a) \quad \forall s, \pi \tag{10}$$

As there are two kinds of algorithms to solve analytically MDP problems, namely: value function iteration and policy value algorithms, we use in this work policy iteration algorithm since it allows to converge faster and better towards the stable policy π^* with less iterations [20]. The most important advantage of this algorithm is that it exploits all transition probabilities thanks to the formula in line 9 that improve of the current policy $\pi_t(s)$ by taking into account the whole state transition matrix. This means that all transitions probabilities are considered at each iteration to expect better the future steps that are required to build the trajectory $\prod^{\pi}$ with the most rewarding actions.

5.4. Q-learning algorithm

As the opposite to MDP, Q-learning uses the bellman equation to update its action value function *Q-value* denoted $Q(s, \pi(s))$. It is an iterative online learning algorithm that does not require the whole transition probabilities to start the learning process. Instead, it requires only the new state at which the current action will lead. In fact, Q-learning is attractive when the state transition probabilities are not known at t_0 when the learning process starts. Nevertheless, according to the Bellman's optimality criterion Eq. (10) [21], there is at least one optimal strategy. Hence, after several iterations, the action values function $Q(s,a)$ is guaranteed to converge to $Q^{\star}$(s,a) [21]. To deal with the complexity of exploring the quality of new actions and exploiting the best explored ones, it combines these two tasks with probability α for the first task (exploration) and $\alpha - 1$ for the second task (exploitation). After several experimental studies, we set α to 0.8 as it is the rate with which Q-learning exploits better previously explored actions.

$$Q_{t+1}(s,a) \leftarrow (1-\alpha) \cdot Q_t(s,a) + \alpha\left(r_{t+1} + \gamma \cdot \max_{a'} Q_t(s',a')\right) \tag{11}$$

As shown in Algorithm 2, at each transmission, a device with an uplink state s, will try to increase its data rate by selecting action a. To update its policy, it observes the reward R and jumps to the observed new state s_{t+1}. Rather than carrying about all possible trajectories in each state like MDP do, it proceeds step by step and tries to build the best trajectory knowing only the next state at which the current action will lead. This makes it slower than MDP since it has to build the state transition probabilities at the same time as the optimal trajectory π^*. For this reason, to exploit the power of MDP, we aim in the next section to build the state transition probabilities based on the quality patterns of LoRa transmission settings.

Algorithm 2: Q-learning algorithm based on FCM.

1: $Q(s,a) \leftarrow 0, a_{init}, s_{init}$
2: $Q(s,a)$
3: $a \leftarrow a_{init}, s \leftarrow s_{init}$
4: **while** True **do**
5: $\quad R[s,a] \leftarrow \Delta U_t$ (Eq. (6))
6: $\quad s' \leftarrow$ argmax P[a] (From FCM)
7: $\quad Q[s,a] \leftarrow$ (Eq. (11))
8: $\quad a \leftarrow \arg\max_{a'} Q(a')$ (Eq. (10))
9: $\quad s \leftarrow s'$
10: **end while**

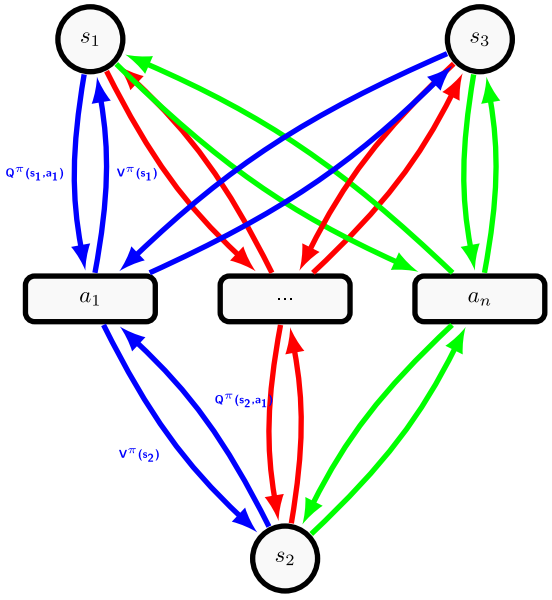


Fig. 5. Markov decision process with n transmission settings and three uplink states.

5.5. Transition matrix P

Both Q-learning and MDP algorithms require an additional knowledge of the wireless environment to know at which state each transmission setting most probably leads. When Q-learning needs only the next state to jump, MDP requires the whole state transition probabilities to switch from one state to another through an action. To take advantage of the power of pattern recognition algorithms to build useful knowledges, we initialize the state transition probabilities P in MDP with the membership degrees values of FCM algorithm. Thus, We consider each resulting cluster/pattern as the aggregation of transmission settings that lead to the same uplink state.

For example, Fig. 5 shows all state transitions between the three uplink states when performing each action. First, we need to distinguish three main quality levels from the quality of all transmission settings. To do that, clustering algorithms are the best candidates to cluster the quality of all transmission settings to three main quality levels. Thus, end-devices that require an uplink state equivalent to s_2 for example, will try to find the path that leads to the state by exploiting more the transmission settings that belong to cluster 2 (see Fig. 5). When a transmission setting a_1 is selected at stat s_x, the probability to jump to s'_1, s'_2 and s'_3 will be recognized through the membership degrees of setting a_1 to the clusters c_1, c_2 and c_3. The advantage of fuzzy clustering compared to hard clustering is its ability to generate membership degrees of each setting to each cluster. Hence, the transition matrix could be built using these membership degrees which disclose the probability that an action could lead to each state. Section below describes in details how we build this matrix.

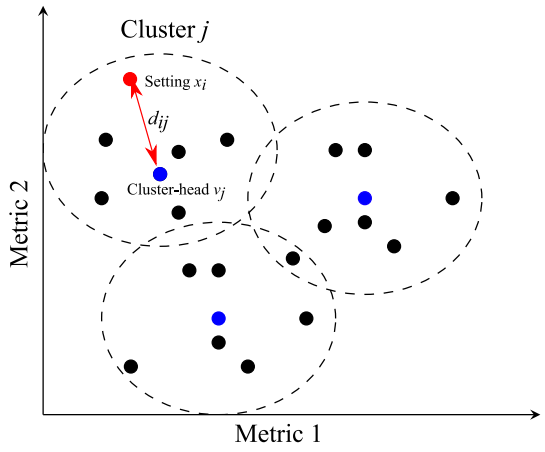


Fig. 6. Clustering of network settings in feature spaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$[\mathbf{P}] = \begin{array}{c} \\ \textbf{action 1} \\ \textbf{action 2} \\ \vdots \\ \textbf{action n} \end{array} \begin{array}{c} \begin{array}{ccc} \textbf{state 1} & \dots & \textbf{state c} \end{array} \\ \begin{pmatrix} m_{11} & \dots & m_{1c} \\ m_{21} & \dots & m_{2c} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{nc} \end{pmatrix} \end{array}$$

6. Initialization of the state transition matrix with the Fuzzy C-Means (FCM) clustering process.

To build a prior knowledge about the quality of LoRa transmission settings, we proposed in a recent work [22] a pattern recognition mechanism by clustering LoRa transmission settings based on their measured QoS metrics. Our goal will be then to map a set of LoRa transmission settings that offer the same QoS to the same cluster. Thus, we propose to use FCM which is an unsupervised clustering algorithm [3] for feature analysis. Unlike hard clustering algorithms, fuzzy clustering algorithms are able to map transmission settings to multiple clusters. For example, a transmission setting with a high SNR and a high ToA can belong to the cluster with the highest reliability constraints since it offers a high SNR. However, it can also belong to another cluster with lower quality constraints since it offers a high ToA. In our context, transmission settings are considered good or bad to a certain degree regarding their membership degrees to different clusters. Hence, instead of mapping this transmission setting to the first cluster [cluster1 = 1] and not the second one [cluster2 = 0], it will belong to both of them with a certain degree [cluster1 = 0.6] and [cluster2 = 0.4]. Such knowledge is mandatory in our study to build a prior knowledge about the consequence of selecting any transmission setting on the quality of uplink transmissions. The clustering is then achieved by minimizing a cost function that depends on the distance between the cluster-heads and the transmission settings.

6.1. Objective function:

The objective of the FCM algorithm is to find a set of membership values M and a set of cluster-heads H that minimize the objective function F in:

$$\min_{(\mathbf{M},\mathbf{H})} \left\{ F_f(\mathbf{M},\mathbf{H}) = \sum_{j=1}^{c} \sum_{i=1}^{n} \mathbf{m}_{ij}^{f} \cdot d_{ij}^{2} \right\} \tag{12}$$

Such that:

$$\text{Constraint:} \quad \sum_{j=1}^{c} \mathbf{m}_{ij} = 1, \forall i \tag{13}$$

$$\text{Distance:} \quad d_{ij}^{2} = \left\| \mathbf{x}_i - \mathbf{h}_j \right\|^2 \tag{14}$$

$$\text{Fuzzification degree:} \quad f > 1 \tag{15}$$

Let p be the number of QoS metrics (features). Let n be the number of all LoRa transmission settings (points). $X = [x_{11}, \dots, x_{np}]$ is a set of p measured QoS metrics of n settings with $x_{ik} \in \mathbb{N}, 1 \le k \le p, 1 \le i \le n$.

The FCM algorithm takes as input a set of metrics X and generates two sets: H and M. $H = [h_{11}, \dots, h_{cp}]$ is a set of cluster heads of p metrics and c clusters with $h_{jk} \in \mathbb{N}$. $M = [m_{11}, \dots, m_{nc}]$ is a set of membership values of n settings to c clusters with $m_{ij} \in \mathbb{R}, 1 \le j \le c$.

6.2. Membership degrees M:

We use this membership matrix in MDP as a state transition matrix P. Thus, transmission settings will lead to different states based on their membership degrees to different clusters. This means that when we pickup one setting from a cluster, FCM is able to recognize at which link quality level the uplink state will jump if we select such an action.

$$[\mathbf{M}] = \begin{array}{c} \\ \textbf{setting 1} \\ \vdots \\ \textbf{setting n} \end{array} \begin{array}{c} \begin{array}{ccc} \textbf{cluster 1} & \dots & \textbf{cluster c} \end{array} \\ \begin{pmatrix} m_{11} & \dots & m_{1c} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{nc} \end{pmatrix} \end{array}$$

The membership-degrees of transmission settings to clusters is inversely proportional to the distance between these settings and the cluster-heads (blue points in Fig. 6). In other words, when a transmission setting get a high membership degree to one cluster, it means that it is close to the clusterhead of this cluster. We use the Eq. (16) to get the membership values of each setting to different clusters.

$$\mathbf{m}_{ij} = \left[\sum_{j'=1}^{c} \left(\frac{d_{ij}}{d_{ij'}} \right)^{\frac{2}{f-1}} \right]^{-1}, \forall j, i \tag{16}$$

6.3. Cluster-heads H:

The cluster-head matrix H is a vector of the measured metrics that are close to all measured metrics in the same cluster and are updated using Eq. (17). At the beginning of the process, the clusterheads (blue point in Fig. 6) are generated randomly. At each iteration, the distance between these blue points and black points (transmission settings) coordinates is updated to decrease the distance between them. After a number of iterations, the clusterheads will be located at the gravity center of each cluster to reduce as much as possible their mean distance to transmission settings coordinates that belong to the same cluster. Once these distances remain stable which mean that the clusterheads are stable, the algorithm converge and the clustering process is stopped. The final distance between the clusterheads and the transmission settings coordinates (Received Signal Strength Indicator (RSSI), ToA, SNR, .etc.) reflects the membership of each setting to the clusters.

$$\mathbf{h}_j = \left[\frac{\sum_{i=1}^{n} \mathbf{m}_{ij}^{f} \cdot \mathbf{x}_i}{\sum_{i=1}^{n} \mathbf{m}_{ij}^{f}} \right], \forall j \tag{17}$$

Algorithm 3 summarizes all the steps described in this section to get the membership degrees' values (M) and cluster heads (H). As reported in [22], the FCM clustering algorithm is able to cluster all possible settings for the three expected clusters. Furthermore, after the convergence of the FCM algorithm, the settings will be ranked based on their membership degrees. This allows the network server to assign the best settings to end-devices that require an uplink with a high quality.

Algorithm 3: Fuzzy C-Means (FCM) clustering algorithm.

1: $X = [x_{11}, \ldots, x_{np}]$
2: $(\mathbf{M}, \mathbf{H})$
3: $t = 0$
4: **while** $F_m(\mathbf{M}_t, \mathbf{H}_t) > \epsilon$ **do**
5: $t = t + 1$
6: Update $\mathbf{H}_t$ from Eq. (17)
7: Update $\mathbf{M}_t$ from Eq. (16)
8: **end while**
9: $(\mathbf{M}, \mathbf{H}) = (\mathbf{M}_t, \mathbf{H}_t)$

Table 2
Simulation settings.

Parameters	Values
Path loss exponent (α)	2.7 (sub-urban environment)
# uplink channels	1
# downlink channels	1
# pkt sent by ED	50
Capture Effect	6.0 dB
Scenarios	
# BS/GW	[1, …, 10]
# ED	$[100, \ldots, 10^3, 2.10^3, \ldots, 10.10^3]$
Packet Size	[10,40,70,100] B
Packet Rate	1 packet per [1,2, …,10] mn
Transmission settings	
Bandwidth	[125, 250, 500] kHz
	[2,5,8,11,14] dBm
Coding Rate	[1,2,3,4]
Spreading Factor	[7,8,9,10,11,12]
Carrier Frequency	868.1 MHz

7. Simulation settings

To evaluate our two algorithms based on FCM clustering, we used datasets generated by LoRaSim simulator [23]. Many parameters related to the environment and the case study were varied to measure the performance of our proposal under different constraints. Transmission parameters including the BW, the CR, the and the SF are tuned by the learning process and adjusted automatically to fit the scenario under study. All these parameters and others are described in Table 2. The path loss exponent is set to 2.7 to reflect the spectral noise in sub-urban environment. As we propose an iterative approach to update transmission settings, we used two channels to exchange data in the uplink channel and acknowledgments in the downlink channel. In addition, more than 28 scenarios have been tested to study the behavior of the network by measuring different Quality of Service (QoS) metrics such as the Data Rate (DR), the Packet Delivery Ratio (PDR), the Time on Air (ToA) and the Transmission Energy (E^{tx}). For this reason, we variate the number of EDs from 100 to 10000, the number of BSs from 1 to 10, the PS from 10 B to 100 B and the PR from one packet per min to one packet per 10 min.

8. Simulation results

In this section, we measure the efficiency of using FCM to disclose at which uplink state transmission settings could lead by comparing the performance of MDP with other algorithms like: Q-learning, EXP3, EXPLoRaTS and ADR that use partially or do not use at all the pattern recognition outputs of FCM. Random algorithm has been included based on the uniform random selection of transmission parameters. This section is divided to three subsections. In the first subsection, we measure the DR and PDR in many scenarios with different numbers of EDs and BSs and different PSs and PRs. We measure the data rate offered by each algorithm in each scenario to know in which scenarios algorithms give a better data rate. In the second subsection, we select one scenario with one BS and 100 EDs that send one packet of 100 B each 4 min. We extend our measured metrics by adding the measured ToA and E^{tx} in addition to PDR and DR to study the time convergence of algorithms. In the third subsection, we select another scenario with 4 BSs and 1000 EDs that send one packet of 70 B each 4 min and we study the same metrics as scenario one.

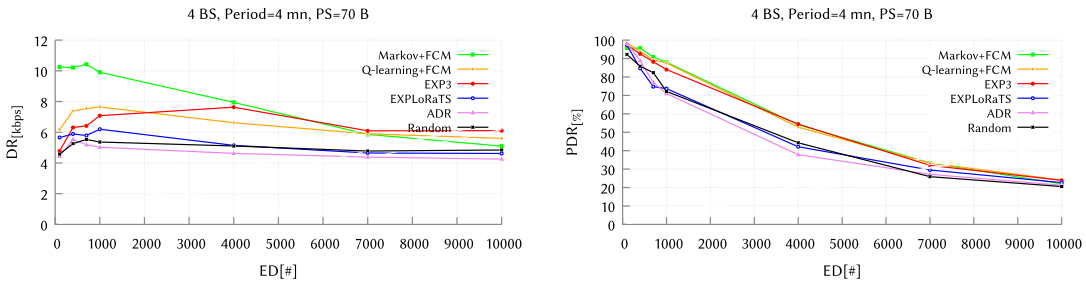
8.1. Measurements of PDR and DR in different scenarios

The main advantage of machine learning algorithms is their ability to learn how the environment behave in each scenario. They are able to fit the scenario under study and converge analytically to the optimal set of actions with few assumptions about the environment. To exploit this advantage, we give in this section an overview of the DR and the PDR of all algorithms in all possible scenarios. Particularly, we highlight through Figs. 7, 8, 9 and 10 the fact that pattern recognition process helped significantly MDP to outperform other solutions.

To measure the scalability of MDP and other algorithms, we study in Fig. 7(a) and Fig. 7(b) the impact of the number of devices on the DR and the PDR of the network. In Fig. 7(b), we observe a decline of the PDR by increasing the number of EDs. However, RL algorithms: Markov, Q-learning and EXP3, always offer a better PDR than other algorithms whatever the number of EDs. As the access to the channel is uniform for all EDs, the advantage of using RL algorithms to increase the uplink traffic is their ability to select transmission settings with a low probability of collisions. Among RL algorithms, MDP which has an additional overall view of all possible state transactions that could happen during the training process, offers the best DR whatever the number of EDs. This result is mainly due to the prior knowledge acquired using FCM before starting the learning process. In the other side, Q-learning has less knowledge about the quality of its actions, since it has to build this knowledge during the learning process from scratch and this slightly decreases its performance.

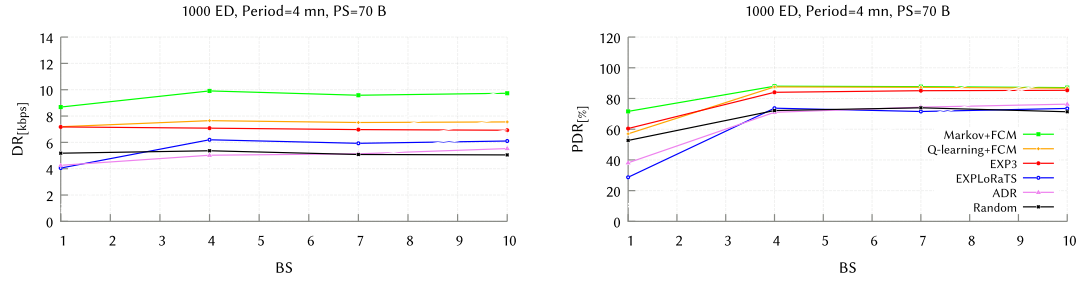
The average data rate of all transmitted packets including the dropped ones is presented in Fig. 7(a). As the number of EDs increases, the DR reaches its highest value when the number of end-devices is less than 1000 for all algorithms except EXP3 since it is the only RL algorithm than does not take advantage of FCM clustering outputs. Thus, it requires more traffic to update its policy. The DR of Markov is the highest one between 10 kbps and 6.5 kbps when the number of ED is lower than 6500. It decreases gradually when we increase the number of EDs due to physical limitations to access the channel. Q-learning is in the second position between Markov and EXP3 algorithms when the number of EDs is lower than 3000. EXP3, in its turn, fails to maximize the DR when the PDR still higher than 50% (see Fig. 7(b)). Thus, it cannot be used in real deployment since most IoT applications require at least a PDR higher than 70%.

We compare in Fig. 8(a) and Fig. 8(b) the impact of the number of cells on DR and PDR, respectively. As the ISM band is very tight especially in Europe, it is mandatory to analyze the impact of the number of BSs on the performance of our algorithms since we use one channel for uplink whether the network is private or public. For this reason, Fig. 8(b) shows an increase of PDR when we increase the number of BSs. This increase of PDR is due to the decrease of the distance between end-devices and BSs by deploying new BSs. Thus, devices will be able to use transmission settings with lower SF and higher BW to send their data to the closest BS, reducing this way the interference between cells and enhancing the overall data rate. However, we can conclude from both Fig. 8(b) and Fig. 8(a) that it is not useful to increase the number of BS above 4 since the PDR and the DR remain the same above this number. With this result, we can reduce the cost of the network installation by purchasing only the necessary number of BSs. When we look at the PDR and the DR of RL algorithms, we see that they offer a higher PDR whatever the number of BSs. Indeed, Markov is the best algorithm to consider with all topologies especially when there is less than 4 cells (see Fig. 8(b)) since it has a



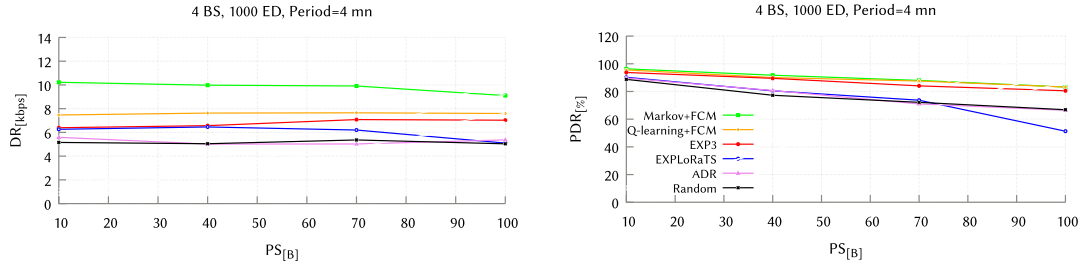
(a). DR vs number of End Devices. **(b).** PDR vs number of End Devices.

Fig. 7. Impact of the number of End Devices on PDR and DR.



(a). DR vs number of Base Stations. **(b).** PDR vs number of Base Stations.

Fig. 8. Impact of the number of Base Stations on PDR and DR.



(a). DR vs Packet Sizes. **(b).** PDR vs Packet Sizes.

Fig. 9. Impact of Packet Sizes on PDR and DR.

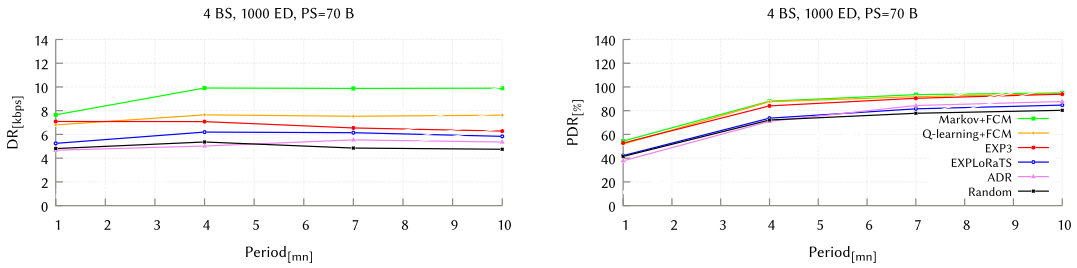
prior knowledge about the quality of each transmission setting through the clustering process.

In a scenario with 4 base stations and 1000 EDs, the DR in Fig. 9(a) remains stable whatever the PS between 10 B and 100 B. However, if we look at the difference of DR between algorithms, we see that MDP offers a higher DR compared to another RL algorithms while keeping the PDR relatively the same. Since there is 4 cells, end-devices are able to use lower SFs to send their packets to the closest BS without interfering with other transmissions in other cells. This decreases considerably the probability of collision. Thus, the PDR in Fig. 9(b) is always higher than 80% whatever the packet size between 10 B and 100 B for RL algorithms.

Depending on IoT applications, IoT devices need to send packets with different sizes that can be high in the case of multimedia data transmission or small to transmit warning alerts. In this context, we study in Fig. 9(a) and Fig. 9(b) the impact of PS on DR and PDR, respectively. As EXPLoRaTS is the only non-iterative algorithm, which means that transmission settings are not updated over time depending on other transmissions, its PDR decreases drastically when the PS becomes higher than 70B. In addition, even if LoRaWAN alliance specifications recommend to use short packets, the PDR and the DR of ADR algorithm remain lower than the PDR and the DR of RL algorithms. Such a result is due to the fact that ADR tries to maximize the DR of each device caring only about the RSSI of recent received packets. Hence, if two devices should use the same SF to increase their DR, none of them could reach the gateway since a collision would happen each time they send a packet at the same time.

Depending on packet transmission rates PRs by devices, many transmission settings with a duty cycle higher than 1% are not allowed. For example, if the PS is equal to 70 B and the SF is equal to 12, the ToA of the transmitted packet is around 2.3s. After which, due to the duty-cycle of 1%, a node has to to remain silent for around 228s ($2.3 * 99$ a little less than 4 min) to send the next frame. Thus, all scenarios with period less than 4 min in this case, are not allowed by LoRaWAN Alliance. In this context, Fig. 10(a) and Fig. 10(b) highlight the impact of packet rates on DR and PDR using one channel for uplink and one channel for downlink. After the analysis of Fig. 7(b) and Fig. 9(b). we fixed the PS and the number of EDs to 70 Bytes and 1000, respectively. Then, we decrease the transmission frequency from one packet per min to one packet per 10 min.

As we decrease the PR, the PDR increase whatever the algorithm used in Fig. 10(b). Indeed, when we decrease the PR, the channel becomes less busy and the probability of collisions decreases significantly. The PDR of RL algorithms still higher than the PDR of all other algorithms since they are able to learn by themselves how to increase the utilization of the network. However, the DR in Fig. 10(a) remains



(a). DR vs Packet Rates. **(b).** PDR vs Packet Rates.

Fig. 10. Impact of Packet Rates on PDR and DR.

relatively stable for all algorithms except for Markov that enhances the DR from 8 kbps to 10 kbps. In fact, Markov with FCM knowledge offers a higher DR whatever the PR between 1 min and 10 min while offering a higher PDR than other algorithms. Q-learning offers less DR (7 kbps) compared to Markov since it does not take advantage of the whole transition matrix to update its Q-values initialized to zero. EXP3 appears to be sensitive to the frequency of transmission since there is a slight decrease of its DR (6 kbps) by decreasing the PR up to one packet per 10 min.

8.2. QoS metrics assessment in the first scenario

The process of finding the optimal reconfiguration policy requires a number of iterations *i.e.* packets exchange. Both MDP and Q-learning algorithms update their action value functions $Q(s,a)$ and converge to the optimal one $Q^{\star}(s,a)$ after a number of state transitions. Whereas, MAB algorithms update their action values $Q(a)$ based only on the number of times the actions have been selected without carrying about the states where they lead. To compare the convergence behavior of all algorithms, we plot in the following figures the measured DR, PDR, ToA and obtained after each transmission during the learning process. We focus in the first scenario on comparing algorithms performances with 100 devices, Then, we increase the number of devices to 1K devices in the second scenario.

Fig. 11(a) shows the comparison of the average DR of the global traffic using MDP, Q-learning, EXP3 ADR, EXPLoRaTS and Random algorithms. When all transmitted packets get a DR higher than 10 kbps using MDP, heuristic algorithms like ADR and EXPLoRaTS offer a DR lower than 5.5 kbps. In fact, in a scenario with 100 devices, there is not enough competition to select transmission settings that belong to the cluster that contains the best settings. So, devices will exploit these settings directly with less probability of collision. In addition, devices will be able to select large BWs since the channel is not fully busy in this scenario. Q-learning algorithm offers the second powerful DR since it uses the knowledge of the clustering process to jump from one state to another based on the membership degrees of each transmission setting to clusters. Random algorithm oscillates without any purpose of convergence since it does not apply any strategy that drive to an optimal data rate.

In Fig. 11(b), we highlight the advantage of using FCM clustering in MDP and Q-learning algorithms to maximize the uplink traffic through the measurement of PDR. As we use in this scenario only 100 devices, heuristic algorithms are able to deal with the low complexity of this scenario and offer a PDR higher than 90%. In fact, as only 100 devices try to access to the channel using ALOHA protocol, the probability of collision is not significant and allows to send more packets. Meanwhile, MDP has the highest data rate since it has all the requested knowledge about the quality of actions and the states where they most probably lead. Q-learning has a slight advantage compared to EXP3 since it has to update its Q-values that are initialized to zero at the beginning of the process. ADR offers a high PDR but less than MDP, Q-learning and EXP3 since it adjusts transmission settings based only on the measured RSSI.

As there is a negative correlation between data rate and time on air, maximizing the data rate will lead to minimize the propagation time since transmission settings with a low SF offer a high data rate and a short time on air. In fact, our solution will try to find a set of transmission settings that have the lowest SF as long as the packets are not dropped. Consequently, selecting the lowest SFs will not only increase the data rate but also mitigate collisions by decreasing the occupancy time of the channel during the transmission. For this reason, MDP in Fig. 12(a) is able to decrease the propagation time of the signal to 0.27 s when all other algorithms offer a ToA higher than 0.4 s.

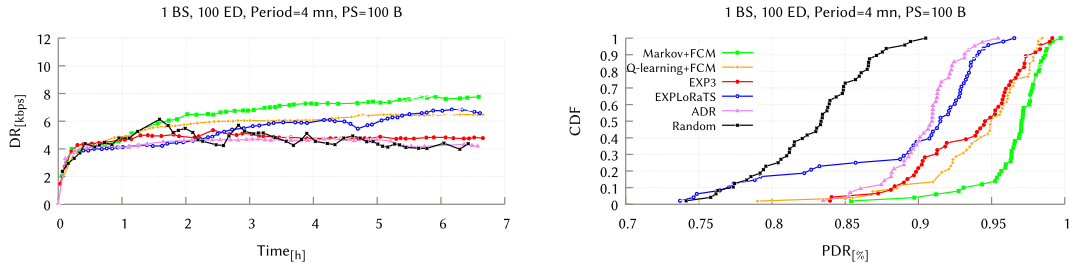
The big advantage of LPWAN networks compared to other wireless technologies is their ability to transmit the signal in a wide area with a low energy consumption. These two properties make LPWAN networks widely used in agricultural industry and by fire fighting services to protect wide forest. For this reason, LoRa devices should optimize their transmission settings to not only increase the uplink traffic but also to increase the life time duration of the network. In this context, our solution mitigate waste of energy by targeting the closest gateway and selecting the lowest SF that can successfully transmit the packets without collision with other transmissions. Hence, MDP is able to mitigate transmissions that waste energy by directly avoiding increasing the P^{tx} and indirectly by avoiding increasing the SF that consume more energy. For this reason, the energy consumed by MDP in Fig. 12(b) during the learning process is lower than the energy consumed by other algorithms.

8.3. QoS metrics assessment in the second scenario

In the second scenario, we increase the number of devices from 100 to 1K devices. As the PDR is at its highest value when the number of BSs is around 4 (see Fig. 8(b)), we select in this scenario 4 BSs that receive packets of 70 bytes each 4 min from each device. Then, we plot the same metrics as scenario one and we compare the same algorithms presented before.

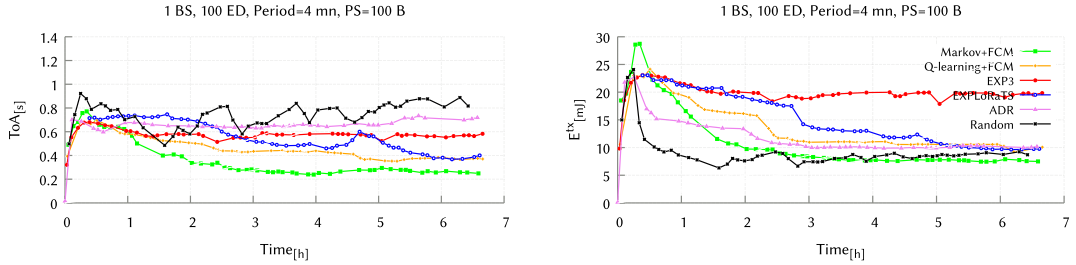
Fig. 13(a) shows the measured average DR of all algorithms over time. Since ADR algorithm has not any knowledge about the data rate obtained after each transmission and observes only the recent measured RSSI, it gives a lower average DR up to 5 kbps (slightly better than Random) compared to RL algorithms and EXPLoRaTS. On the other hand, when using Q-learning and Markov algorithms, the DR is improved significantly to achieve 8 kbps and 9 kbps, respectively. This result is made by the frequent update of transmission settings taking into account the state of the link to balance the load of uplink traffic. Both MDP and MAB algorithms have the ability to explore enough actions (and states with MDP) to be able to exploit the better ones and perform better DR. EXPLoRaTS has a better DR compared to ADR but shows its weakness compared to RL algorithms.

When we look at the PDR in Fig. 13(b), we see that RL algorithms offer a better successful received packet rate compared to EXPLoRaTS, ADR and random algorithms. In fact, RL algorithms are able to enhance



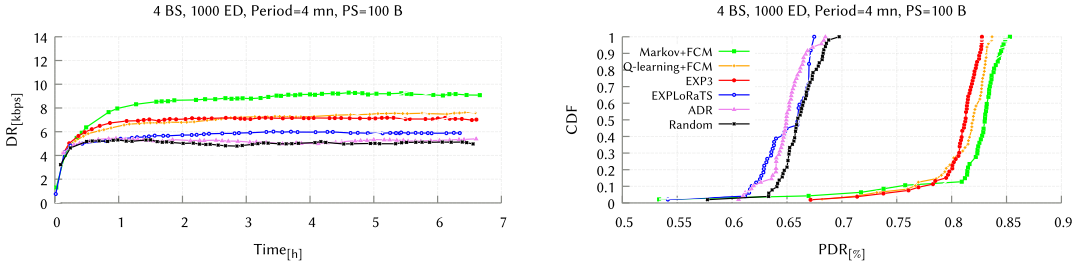
(a). DR of algorithms over time. **(b).** PDR of algorithms over time.

Fig. 11. PDR & data rate in the first scenario.



(a). ToA of algorithms over time. **(b).** E^{tx} of algorithms over time.

Fig. 12. ToA and energy consumption in the first scenario.



(a). DR of algorithms over time. **(b).** PDR of algorithms over time.

Fig. 13. PDR & data rate in the second scenario.

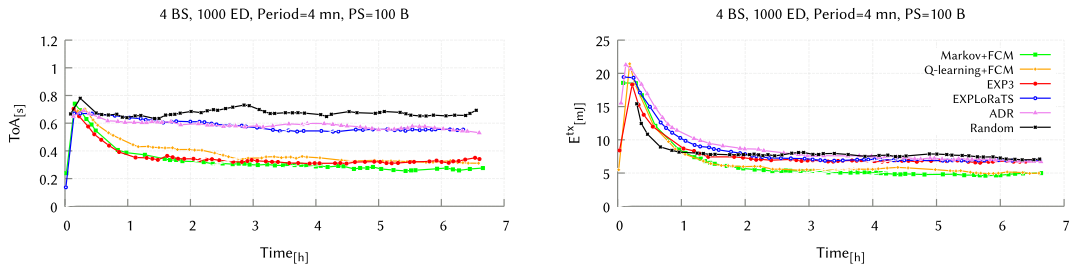
the PDR by 20%; from 65% to 85%. However, among RL algorithms, Markov offers a better PDR thanks to the quality patterns acquired by FCM to disclose the quality of each setting before starting the learning process. Simulations have been carried out for a period of 7 h but Fig. 13(a) shows that almost all algorithms reach their highest DR after only four hours. The main reason for this is the high density of traffic that makes the controller receive enough requests from 1000 devices through 4 BSs to update its policy.

Many IoT applications nowadays become more and more sensitive to the delay of transmissions when it comes to alerting or synchronized systems. Thus, we study in this work the ToA obtained during the training process. Fig. 14(a) shows a decrease of ToA from 0.7 s to 0.35 s when RL algorithms are applied. This means that RL algorithms are able to decrease the transmission delay by 50% compared to random, ADR and EXPLoRaTS algorithms. However, Markov remains the only RL algorithm with the best tradeoff between all QoS metrics including DR, PDR, ToA and in Fig. 14(b). In addition, in less that two hours, more than 70% of the transmitted packets reach the gateway in only 0.3 s using Markov with FCM when almost all transmitted packets with other algorithms have a ToA higher than 0.5 s in average.

In the context of LPWAN, saving devices energy consumption is mandatory to extend the network life time. For this reason, Fig. 14(b) shows the average energy consumption per packet with each algorithm. As the relationship between SF and is inversely proportional, maximizing the DR leads to the selection of lower SFs that consume less energy. Thus, Markov and Q-learning with FCM knowledge are able to reduce the waste of energy better than EXP3 and all other algorithms. They have a better energy efficiency with around 5mJ per packet when other algorithms consume more than 7mJ in average.

9. Conclusion

Long Range Wireless Access Network (LoRaWAN) is among the leading wireless Internet of things (IoT) networks due to its large coverage and low energy consumption. One of the main challenges of this work was to enhance the network quality when new applications' requirements arise. In this work, we addressed the reconfiguration problem of Long Range (LoRa) transceivers' parameters. We proposed a new approach for dynamic reconfiguration using Fuzzy C-Means (FCM) clustering, Markov Decision Process (MDP): Markov and Q-learning. Our main achieved goals are: (i) characterization of transmission parameters based on different Quality of Service (QoS) metrics, (ii) maximization of the network Data Rate by tuning parameters via trial/reward process, (iii) and performance evaluation and comparison with solutions proposed in the literature. Our simulation results show that MDP with FCM clustering allow to improve the DR, Packet Delivery Ratio (PDR), Time on Air (ToA) and Transmission Energy (E^{tx}) of the network in many scenarios with different numbers of End Device (ED)s and Base Station (BS)s. The PDR and the DR were improved by



(a). ToA of algorithms over time. **(b).** E^{tx} of algorithms over time.

Fig. 14. ToA and energy consumption in the second scenario.

25%, the ToA was reduced by 40% and was reduced by 20%. As a future work, we plan to use contextual Multi-Armed Bandit (MAB) algorithms to consider the state of the link when estimating the mean reward of each arm.

CRediT authorship contribution statement

Aghiles Djoudi: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Rafik Zitouni:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Nawel Zangar:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Laurent George:** Conception and design of study, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

All authors approved the version of the manuscript to be published.

References

[1] Tommaso Polonelli, Davide Brunelli, Achille Marzocchi, Luca Benini, Slotted ALOHA on LoRaWAN-design, analysis, and deployment, Sensors (ISSN: 1424-8220) 19 (4) (2019) 838, http://dx.doi.org/10.3390/s19040838, URL: http://www.mdpi.com/1424-8220/19/4/838.

[2] LoRa-Alliance, Lorawan^{TM}1.1 specification, april, 2018, URL: https://lora-alliance.org/lorawan-for-developers.

[3] James C. Bezdek, Robert Ehrlich, William Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. FCM (ISSN: 00983004) 10 (2–3) 191–203, http://dx.doi.org/10.1016/0098-3004(84)90020-7, URL: https://linkinghub.elsevier.com/retrieve/pii/0098300484900207.

[4] Enrique H. Ruspini, James C. Bezdek, James M. Keller, Fuzzy clustering: a historical perspective, IEEE Comput. Intell. Mag. (ISSN: 1556-6048) 14 (1) 45–55, http://dx.doi.org/10.1109/MCI.2018.2881643.

[5] Francesca Cuomo, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, Patrizio Pisani, EXPLoRa: extending the performance of LoRa by suitable spreading factor allocations, in: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, ISBN: 978-1-5386-3839-2, pp. 1–8, http://dx.doi.org/10.1109/WiMOB.2017.8115779, URL: http://ieeexplore.ieee.org/document/8115779/.

[6] Naoki Aihara, Koichi Adachi, Osamu Takyu, Mai Ohta, Takeo Fujii, Q-learning aided resource allocation and environment recognition in LoRaWAN With CSMA/CA, IEEE Access (ISSN: 2169-3536) 7, 152126–152137, http://dx.doi.org/10.1109/ACCESS.2019.2948111, URL: https://ieeexplore.ieee.org/document/8873564/.

[7] Inaam Ilahi, Muhammad Usama, Muhammad Omer Farooq, Muhammad Umar Janjua, Junaid Qadir, LoRaDRL: Deep reinforcement learning based adaptive PHY layer transmission parameters selection for LoRaWAN, in: 2020 IEEE 45th Conference on Local Computer Networks, LCN, IEEE, 2020, http://dx.doi.org/10.1109/lcn48667.2020.9314772.

[8] Yi Yu, Lina Mroueh, Shuo Li, Michel Terre, Multi-agent Q-learning algorithm for dynamic power and rate allocation in LoRa networks, in: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, IEEE, 2020, http://dx.doi.org/10.1109/pimrc48278.2020.9217291.

[9] Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, Steven Martin, LoRa-MAB: a flexible simulator for decentralized learning resource allocation in IoT networks, in: 2019 12th IFIP Wireless and Mobile Networking Conference, WMNC, (ISSN: 2473-3644) pp. 55–62, http://dx.doi.org/10.23919/WMNC.2019.8881393.

[10] Riccardo Marini, Konstantin Mikhaylov, Gianni Pasolini, Chiara Buratti, LoRaWANSim: a flexible simulator for LoRaWAN Networks, Sensors (ISSN: 1424-8220) 21 (3) 695, http://dx.doi.org/10.3390/s21030695, URL: https://www.mdpi.com/1424-8220/21/3/695.

[11] Arshad Farhad, Jae-Young Pyun, HADR: a hybrid adaptive data rate in lorawan for internet of things, ICT Express (ISSN: 24059595) 8 (2) 283–289, http://dx.doi.org/10.1016/j.icte.2021.12.013, URL: https://linkinghub.elsevier.com/retrieve/pii/S2405959521001788.

[12] Farzad Azizi, Benyamin Teymuri, Rojin Aslani, Mehdi Rasti, Jesse Tolvanen, Pedro H.J. Nardelli, MIX-MAB: reinforcement learning-based resource allocation algorithm for LoRaWAN, 2022, http://dx.doi.org/10.48550/arXiv.2206.03401, arXiv:2206.03401.

[13] Dimitrios Zorbas, Khaled Q. Abdelfadeel, Victor Cionca, Dirk Pesch, Brendan O'Flynn, Offline scheduling algorithms for time-slotted lora-based bulk data transmission, in: 2019 IEEE 5th World Forum on Internet of Things, WF-IoT, 2019, pp. 949–954, http://dx.doi.org/10.1109/WF-IoT.2019.8767277.

[14] Dimitrios Zorbas, Christelle Caillouet, Khaled Abdelfadeel Hassan, Dirk Pesch, Optimal data collection time in LoRa networks—a time-slotted approach, Sensors (ISSN: 1424-8220) 21 (4) (2021) 1193, http://dx.doi.org/10.3390/s21041193, URL: https://www.mdpi.com/1424-8220/21/4/1193.

[15] Martin Bor, Utz Roedig, LoRa transmission parameter selection, in: 2017 13th International Conference on Distributed Computing in Sensor Systems, DCOSS, IEEE, ISBN: 978-1-5386-3991-7, pp. 27–34, http://dx.doi.org/10.1109/DCOSS.2017.10, URL: http://ieeexplore.ieee.org/document/8271941/.

[16] Joakim Eriksson, Jonas Skog Andersen, Investigating the Practical Performance of the LoRaWAN Technology, 61.

[17] Semtech, Semtech LoRa Technology Overview. URL: https://www.semtech.com/lora.

[18] Raja Karmakar, Samiran Chattopadhyay, Sandip Chakraborty, Linkcon: adaptive link configuration over SDN controlled wireless access networks, in: Proceedings of the ACM Workshop on Distributed Information Processing in Wireless Networks - DIPWN'17, ACM Press, ISBN: 978-1-4503-5051-8, pp. 1–6, http://dx.doi.org/10.1145/3083181.3083184, URL: http://dl.acm.org/citation.cfm?doid=3083181.3083184.

[19] Herbert Robbins, Some aspects of the sequential design of experiments, Bull. Amer. Math. Soc. 58 (5) (1952) 527–536, http://dx.doi.org/10.1090/s0002-9904-1952-09620-8.

[20] Richard S. Sutton, Andrew G. Barto, Reinforcement learning: An introduction, second ed., in: Adaptive Computation and Machine Learning Series, The MIT Press, ISBN: 978-0-262-03924-6, 2018.

[21] Christopher J.C.H. Watkins, Peter Dayan, Technical note: Q-learning, Machine Learning (ISSN: 1573-0565) 8 (3) (1992) 279–292, http://dx.doi.org/10.1023/A:1022676722315.

[22] A. Djoudi, R. Zitouni, N. Zangar, L. George, Reconfiguration of LoRa networks parameters using fuzzy C-means clustering, in: 2020 International Symposium on Networks, Computers and Communications, ISNCC, pp. 1–6, http://dx.doi.org/10.1109/ISNCC49221.2020.9297284.

[23] Martin C. Bor, Utz Roedig, Thiemo Voigt, Juan M. Alonso, Do LoRa low-power wide-area networks scale? in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16, in: Evaluation, ACM Press, ISBN: 978-1-4503-4502-6, pp. 59–67, http://dx.doi.org/10.1145/2988287.2989163, URL: http://dl.acm.org/citation.cfm?doid=2988287.2989163.