# Access Control in NB-IoT Networks: A Deep Reinforcement Learning Strategy

Yassine Hadjadj-Aoul, Soraya Aït-Chellouche

# Access Control in NB-IoT Networks: A Deep Reinforcement Learning Strategy

Yassine Hadjadj-Aoul
*Univ Rennes, Inria, CNRS, IRISA*
Rennes, France

Soraya Ait-Chellouche
*Univ Rennes, Inria, CNRS, IRISA*
Rennes, France

***Abstract*—The Internet of Things (IoT) is a key enabler of the digital mutation of our society. Driven by various services and applications, Machine Type Communications (MTC) will become an integral part of our daily life, over the next few years. Meeting the ITU-T requirements, in terms of density, battery longevity, coverage, price, and supported mechanisms and functionalities, Cellular IoT, and particularly Narrowband-IoT (NB-IoT), is identified as a promising candidate to handle massive MTC accesses. However, this massive connectivity would pose a huge challenge for network operators in terms of scalability. Indeed, the connection to the network in cellular IoT passes through a random access procedure and a high concentration of IoT devices would, very quickly, lead to a bottleneck. The latter procedure needs, then, to be enhanced as the connectivity would be considerable. With this in mind, we propose, in this paper, to apply the access class barring (ACB) mechanism to regulate the number of devices competing for the access. In order to derive the blocking factor, we formulated the access problem as a Markov decision process that we were able to solve using one of the most advanced deep reinforcement learning techniques. The evaluation of the proposed access control, through simulations, shows the effectiveness of our approach compared to existing approaches such as the adaptive one and the Proportional Integral Derivative (PID) controller. Indeed, it manages to keep the proportion of access attempts close to the optimum, despite the lack of accurate information on the number of access attempts.**

***Index Terms*—cellular IoT, massive access, reinforcement learning, access control, congestion control.**

## I. Introduction

IoT objects connections and especially Machine-to-Machine (M2M) communications are considered as one of the most important evolutions of the Internet. Supporting these devices is, however, one of the most important challenges facing network operators [Lin *et al.(2016)Lin, Adhikary, and Eric Wang*]. Indeed, the huge number of devices that might try to access the network at the same time could lead to heavy congestion or even total saturation, with all the consequences that this may entail. Indeed, as can be seen in Figure 1, a very limited number of devices simultaneously trying to access the network may drop network performance down to zero, regardless of the available access possibilities [Bouzouita *et al.(2016)Bouzouita, Hadjadj-Aoult, Zangar, and Tabbane*]. Under these circumstances, it seems obvious that effective access control mechanisms are needed to maintain a reasonable number of access attempts.

The Third Generation Partnership Project (3GPP) identified the overloading of the random access network (RAN) as a priority at an early stage and proposed several solutions. Among the suggested approaches, the Access Class Barring (ACB), proposed in version 8, and its extension, the Extended Access Barring (EAB), proposed in version 11, are certainly the most effective strategies [3GPP(2011)]. Indeed, these approaches tackle the problem at its root by preventing even the attempts to access the network. However, these approaches only provide a framework for congestion control, without giving a ready-made solution. Indeed, the main idea behind the ACB is to calculate a blocking factor to prevent terminals from accessing the network, but no solution for the calculation of this factor has been proposed.

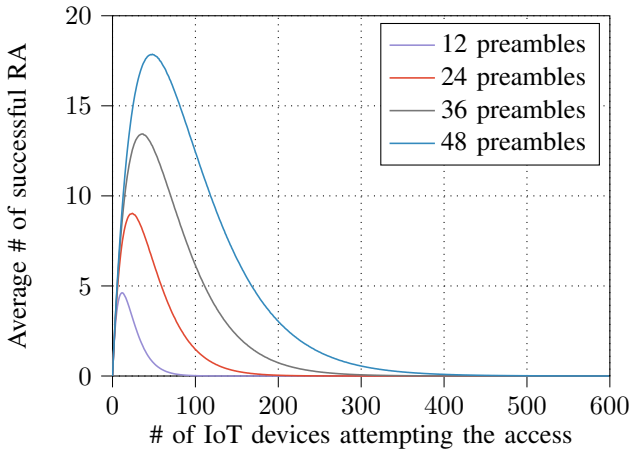



Fig. 1: The average number of access successes as a function of the number of devices, for different access opportunities.

The blocking factor calculation requires a good knowledge of the number of terminals willing to attempt access in order to deduce the optimal blocking probability. This information is unfortunately not available in the network (i.e., the state of the network is not observable). In order to solve this problem, two important challenges must be addressed: (1) estimating the number of devices attempting the access simultaneously, and (2) designing an access control strategy for the dynamic generation of the blocking factor.

Several solutions have been proposed to estimate the number of devices willing to access the network [Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino*]. These estimators are, however, highly noisy and dependent

on the blocking factor. On the other hand, many existing techniques are dependent on a particular traffic pattern, leading to improper actions when the traffic pattern changes. In reality, IoT objects do not follow a single model but a mixture of models making the prediction of this type of traffic very difficult sometimes. IoTs may follow, indeed, different laws at the same time: Poisson (e.g., credit machine in shops), Uniform (e.g., traffic lights), and Beta (e.g., event driven). Therefore, we propose, in this paper, exploiting the potential of the most advanced reinforcement learning techniques in order to take into account this complex reality and deduce a sub-optimal control strategy. More specifically, we exploit the Twin Delayed Deep Deterministic policy gradient algorithm (TD3) [Fujimoto *et al.(2018)Fujimoto, van Hoof, and Meger*] to produce, from past estimates, the optimal blocking factor regardless of the uncertainties on the number of devices willing to access the network.

The different findings of this paper are summarized in the following:

- We propose a detailed and up-to-date state of the art.
- We describe a fluid model of IoT access in NB-IoT networks.
- We formulated the problem of IoT access as a Markov Decision Process (MDP).
- We design a specific reward function in order to guide the agent to improve the quality of the solutions.
- We provide an in-depth analysis of the problem.

The remainder of this paper is organized as follows: Section II provides the fundamentals of the random access procedure, addressed in this paper, and gives an overview of the main techniques for congestion control in IoT networks, with a focus on cellular IoT networks. Section III describes the network access model for IoT terminals. Section IV presents the details of the proposed control solution, based on the TD3 algorithm, adapted to solve the blocking factor calculation problem. Section V presents the simulation environment of the proposed approach and shows its efficiency compared to the existing one. Finally, the paper concludes with a summary of the main advantages and achievements of the proposed system in Section VI.

## II. State of the Art

### A. Random Access Fundamentals

The Narrowband Physical Random Access CHannel (NPRACH) has been completely redesigned in NB-IoT to improve network coverage and power consumption, but also to accommodate the narrowband nature of NB-IoT [Lin *et al.(2016)Lin, Adhikary, and Eric Wang*]. Indeed, in Long-Term Evolution (LTE), the PRACH channel alone occupies more bandwidth than NB-IoT as a whole (1.08 MHz vs. 180 kHz). Each NB-IoT terminal, willing to connect or resynchronize to the base station on its uplink, after a long period of inactivity, should perform a random access procedure. The first step of the latter consists of transmitting a sequence of preambles on one of the frequencies, periodically allocated to the NPRACH channel which is called the Random Access Opportunity (RAO). The preamble consists of a set of four groups of OFDM symbols, as shown in Figure 2. Each group of symbols consists of a cyclic prefix (CP) and a set of data symbols. In order to maintain the orthogonality of random access transmissions on different sub-carriers, the CP must be long enough to compensate for long round-trip times, especially in cells as large as those targeted in NB-IoT (up to 35 km) [3GPP(2015)]. The higher the number of data symbols, the lower the CP overflow. On the other hand, this number should be kept small in order to control interference. In NB-IoT, the number of data symbols is set to 5 and two CP lengths are defined for the two NPRACH channel formats, namely 266.7 s and 66.7 s [Lin *et al.(2016)Lin, Adhikary, and Eric Wang*], [ETSI(2020a)].

For coverage extension purposes, the preamble can be repeated $k$ times $(k = 2^i, i = 0, \cdots, 7)$ [ETSI(2020b)]. The sequence of preambles sent by the terminal therefore consists of $4 \times 2^i$ groups of symbols. The network can thus define up to three different configurations of the NPRACH resource per cell, depending on the considered coverage classes. The number of repetitions $k$ is therefore defined for each configuration.

Each symbol group is modulated on a different subcarrier than the others. The NPRACH channel uses only the single tone mode with a 3.75 kHz spacing. A frequency band of up to 48 subcarriers can then be allocated to this channel with a baseband of 12 subcarriers. Thus, 12, 24, 36, or 48 contiguous subcarriers are allocated to this channel in each coverage class. Therefore, the terminal has 12, 24, 36 or 48 orthogonal preambles and randomly selects one to be transmitted. In addition, as illustrated in Figure 2, NB-IoT defines two hopping patterns in the frequency band allocated to the NPRACH: (i) a fixed pattern for hops between different symbol groups constituting the same preamble and (ii) a pseudo-random pattern for the preamble repetitions. Thus, within a preamble, a jump of one subcarrier is applied between the first and the second group of symbols and between the third and the fourth group of symbols. Another hop of six subcarriers is also applied between the second and third group of symbols. A pseudo-random model, based on the cell identifier and the expected number of repetitions, is applied to choose the sub-carrier indexes at the beginning of the different repetitions of the preamble [Lin *et al.(2016)Lin, Adhikary, and Eric Wang*].

The random access procedure in NB-IoT, illustrated in Figure 3, is a four-step exchange between the terminal and the base station [ETSI(2019)]:

- The terminal transmits the selected preamble at the first RAO and sets a timer to receive the Random Access Response (RAR);
- If the preamble is well detected by the base station, the base station sends a RAR response carrying the synchronization advance and the allocated resource;
- The base station executes the contention resolution and sends the identity of the winning terminal in the contention resolution message. If the message doesn't arrive

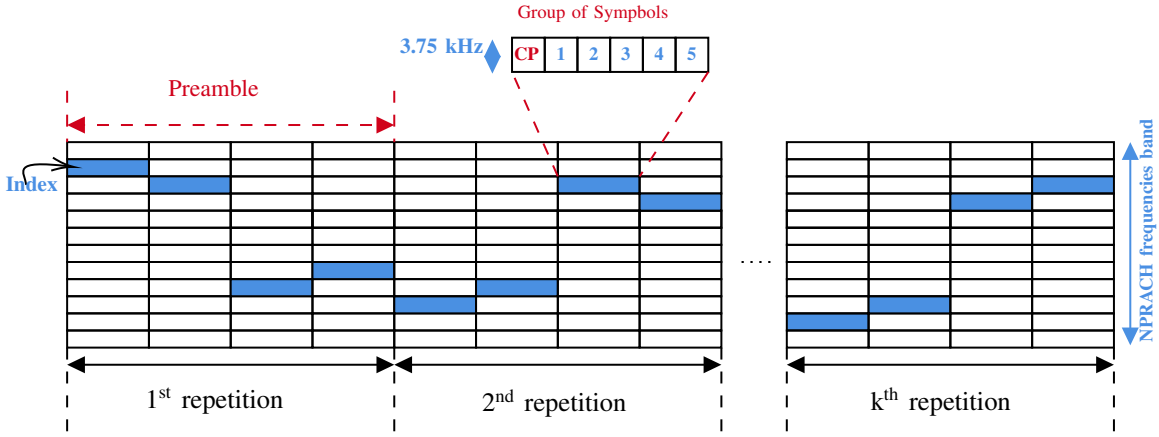



Fig. 2: Preamble sequence structure.

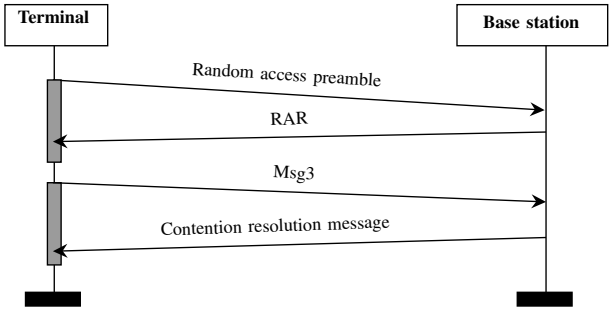



Fig. 3: Random access procedure.

at the terminal side, the terminal continues waiting until the timer expires;

- The terminal then sends a connection request, using the resource allocated to it, and re-arms a contention resolution timer. This request, named msg3, carries the identity of the terminal.

This procedure fails if the terminal doesn't receive one of the two responses from the base station within the time windows defined by the two timers. Collisions between preambles, sent by different terminals, are often the cause of the failure. Indeed, if two or more terminals choose the same preamble on the same ROA, each of their access attempt fails.

Each terminal, for which the access procedure has failed, observes a waiting time chosen uniformly and at random within a predefined interval, and then retransmits its preamble. The number of allowed retransmissions depends on the coverage class of the terminal. If this number is reached and the terminal still doesn't pass the access procedure, the terminal moves to the higher coverage class, if the latter is configured, or concludes to the definitive failure of the access procedure.

### *B. Related Work*

Beyond the broadband generalization, 5G promises to improve our daily life through connected ecosystems. Indeed, 5G, via Massive Machine-Type Communications (mMTC), goes a lot further by enabling seamless and massive connectivity of things. If this vision of IoT seems very attractive, it also drives huge challenges, especially from the resource management point of view. Network operators have imperatively to scale their IoT networks in order to efficiently manage this excessively large number of sensors that should be connected, in the coming years.

As explained in Section II-A, IoT terminals connecting to the network should first initiate a random access procedure. However, the latter was initially designed for a limited number of terminals and the high density targeted by NB-IoT, and mMTC in general, can very quickly lead to a severe congestion. Indeed, the number of preambles available at each RAO being limited, the greater the number of terminals attempting to access the network, the greater the risk of collision, thus leading to the failure of the procedure for all the terminals having chosen the same preamble. The terminals that fail their access attempt can retransmit the preamble after observing a backoff time, but these retransmissions can also lead, on the one hand, to a spectral resource wasting, and, on the other hand, to increase energy consumption at the terminal level [Harwahyu *et al.(2019)Harwahyu, Cheng, Tsai, Hwang, and Bianchi*].

Being a critical phase, the random access procedure has been the subject of numerous academic studies. Some of them, such as [Baracat and Brito(2018)], [Jiang *et al.(2018)Jiang, Deng, Condoluci, Guo, Nallanathan, and Dohler*], [Harwahyu *et al.(2018)Harwahyu, Cheng, Wei, and Sari*], have proposed analytical models for optimizing the success probability of the access attempts and the average access time in different network configurations and, in particular, under time constraints [Harwahyu *et al.(2018)Harwahyu, Cheng, Wei, and Sari*]. Other works have focused on retransmissions. Thus, a Markov chain-based model was proposed to model the number of retransmissions in [Harwahyu *et al.(2019)Harwahyu, Cheng, Tsai, Hwang, and Bianchi*], [Sun *et al.(2018)Sun, Tong, Zhang, and He*], and the authors proposed a model to find a trade-off between the number of repetitions planned

in the physical layer and the number of retransmissions planned in the MAC layer and optimize these two values basing on a target successful probability. The study concluded that retransmissions considered in NPRACH can reduce the number of repetitions. These latter are only required under worse channel conditions.

In [Lin *et al.(2016)Lin, Adhikary, and Eric Wang*], [Jeon *et al.(2018)Jeon, Seo, and Jeong*], [Hwang *et al.(2019)Hwang, Li, and Ma*], the focus was on the transmission of the preamble and the estimation of the Time of Arrival (TA). Thus, a receiver-side detection algorithm, a new NPRACH frequency domain hopping model, and a Framework for the detection of multiple users have been proposed, respectively. In [Zhang *et al.(5555)Zhang, Xie, and Wang*], the TA of preambles that has suffered a collision is used to improve the performance of the random access procedure.

From the standardization side, congestion control at the network access level was identified early as a priority by the 3GPP and ETSI organizations [3GPP(2011)]. The cellular IoT, and particularly NB-IoT, therefore naturally inherit existing solutions. Among these solutions, we can find the ACB (Access Class Barring) mechanism and its extension EAB (Extended Access Barring), slotted random access, MTC-specific backoffs, dynamic allocation of resources, etc. [Ali *et al.(2017)Ali, Hossain, and Kim*].

The ACB and EAB mechanisms are the ones that tackle the problem at its roots by blocking access to the network. The blocking parameters, namely the blocking probability $p$ and a blocking time $T_b$, are broadcasted in the System Information Block (SIB) at each RAO. Each terminal attempting to access the network generates an access probability $q$. If $q < p$, the terminal has permission to make its access attempt; otherwise, the latter is deferred for a time $T_b$. In the EAB extension, the terminals are further classified within different priorities according to their QoS requirements and the EAB algorithm dynamically blocks low priority terminals based on the arrival rate by broadcasting a bitmap in SIB-14.

It seems clear that congestion control based on these mechanisms relies, entirely, on the blocking probability defined by the network. Indeed, if the blocking probability is too high, then a large number of terminals would pass the access control, thus leading to collisions and if, on the other hand, this probability is too small then, the collisions would be reduced, but a large number of terminals would switch to idle mode and this would lead to underused resources. Thus, it is essential to optimize the blocking probability to efficiently control the congestion at the access level.

A study of the performance of ACB and EAB was conducted in [Toor and Jin(2017)]. The comparison of the two techniques through simulation has shown that ACB is more suited to high delay constrained communications and EAB performs better in the case of energy-constrained terminals. However, the optimization of the blocking factor requires knowing the number of terminals willing to access the network at the base station level, which, in practice, is not the case. In fact, the base station doesn't know the number of terminals whose access attempts have been blocked.

Several mechanisms have been proposed to estimate the number of terminals attempting to access the network (including terminals blocked by the access control) in order to derive the blocking factor to be used. In [Park and Lim(2016)], considering the number of blocked terminals not known, the authors use a heuristic to adapt the probability of blocking. In [Liu *et al.(2020)Liu, Agiwal, Qu, and Jin*], the proposed algorithm is based on a recursive Bayesian estimate of the active terminals in each class and based on this, preambles are allocated to the different classes. The algorithm was then improved by assigning an ACB blocking factor to each class, independently of the others, for better congestion control. In [Jin *et al.(2017)Jin, Toor, Jung, and Seo*], a recursive Bayesian estimate of active terminals, based on the number of unselected preambles, allows for calculating a blocking factor for sporadic terminal arrivals. In [Cheng *et al.(2015)Cheng, Chen, Chen, and Wei*], the performance of the EAB mechanism is studied for LTE-A networks. The optimal values of the paging cycle as well as the periodicity of SIB14 are then derived by an analytical model subjected to a targeted QoS constraints.

In this paper, we use on an estimator proposed in a previous work [Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino*] and, unlike the works cited above, we use reinforcement learning techniques, namely the TD3 algorithm, to derive an optimal blocking probability from a set of past estimates (an estimates' horizon). To our knowledge, this is the first time that the latter algorithm has been used in the management of massive accesses in NB-IoT networks.

## III. A Fluid Model for IoT Device Access

The proposed model, which was firstly introduced in [Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino*], provides an overview of IoT devices executing the ACB algorithm. We consider in this work a simplification since we consider only one class of service.

During the random access attempt, the IoT devices compete for the same available preambles. As stated in the 3GPP standard, the number of preambles $N$ must be an integer in the set $\{12, 24, 36, 48\}$ [Agiwal *et al.(2019)Agiwal, Maheshwari, and Jin*].

In each Random Access CHannel (RACH) opportunity, these preambles are split into successful (i.e., chosen by only one device), collided (i.e., chosen by two or more devices) and idle (i.e., selected by none of the devices) preambles. In the following, we compute the average values of these quantities that we have determined in [Bouzouita *et al.(2015)Bouzouita, Hadjadj-Aoul, Zangar, Rubino, and Tabbane*]. These quantities will be used by our algorithms.

Let's define $q_N = 1 - 1/N$. The average number of successful preambles $N_S$, during the RACH opportunities, is given as follows (this is a classic "balls into bins" problem):

$$N_S = q_N^{x_2 - 1} x_2 \tag{1}$$

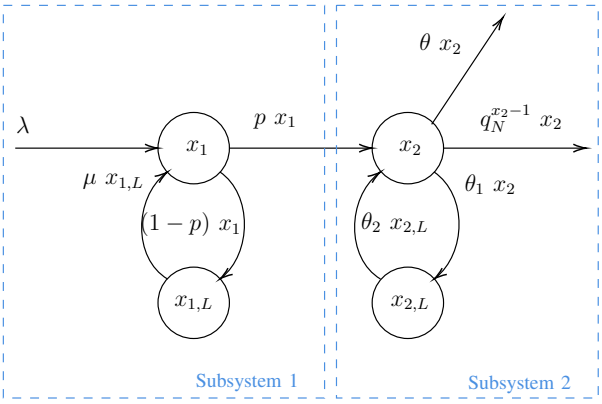



Fig. 4: System model. Subsystem 1 represents the terminals that would like to connect; the objects in the state variable $x_1$ represent those that can try to connect with a probability $p$, in the case of a failure they go into the waiting state $x_{1,L}$ for a back-off time duration. Subsystem 2 represents the objects coming from the different classes that can try to choose a preamble. In the case of a collision, they may attempt access a number of times. They leave subsystem 2 when they succeed in being the only ones to have chosen a preamble or when they reach the maximum number of attempts (with a rate of $\theta$).

where $x_2$ represents the number of devices attempting the access. As demonstrated in [Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino*], Equation (1) is maximized (i.e., derivative equals to zero) when the number of devices $x_2^*$ attempting access simultaneously is equal to

$$x_2^* = -\frac{1}{\ln q_N} \tag{2}$$

Knowing the optimal number of devices accessing the network, one can obtain the corresponding number of successes $N_S^*$, which can be given by

$$N_S^* = -\frac{1}{eq_N \ln q_N} \tag{3}$$

The average number of idle preambles $N_I$ is given by the following equation:

$$N_I = Nq_N^{x_2}. \tag{4}$$

From (1) and (4), we obtain the expected number of failed preambles $N_F$:

$$N_F = N - (N_S + N_I). \tag{5}$$

The modeled system is an approximation of reality in many ways, in particular with regard to the limited and fixed number of access attempts. However, we preferred to simplify the model to make it more tractable (see Figure 4). Furthermore, a system where devices often reach the maximum number of attempts is an unstable system, which we are naturally trying to avoid. On the other hand, the Coverage Enhancement (CE) strategy, as introduced in NB-IoT, is not considered in this work, although it will be investigated in future work.

The proposed model is a fluid one: the involved quantities and the whole numbers are seen as real (continuous) quantities. The parameters used are listed below:

- $x_1(t)$ is the number of backlogged devices at time $t$;
- $x_{1,L}(t)$ is the number of blocked devices waiting for a re-attempt at time $t$, after having failed an ACB check;
- $x_2(t)$ is the total number of devices from the different classes that pass the ACB check and wait to start Random Access (RA) attempt at time $t$;
- $x_{2,L}(t)$ is the number of blocked devices at time $t$ after a failed RA attempt and waiting to try again;
- $\lambda$ is the arrival rate of devices. Different traffic patterns could be considered, depending on the type of IoT applications;
- $\mu$ is the rate of ACB re-attempts;
- $\theta_1$ is the rate of RA failure, which is equal to $1 - q_N^{x_2-1}$ when $\theta$ is equal to 0 (see last item);
- $\theta_2$ is the rate of RA re-attempts;
- $\theta$ is the rate at which the devices abort the transmission after reaching the maximum number of RA attempts; in a correctly dimensioned system, we should have $\theta = 0$;
- $p$ is the ACB factor.

Now, we are ready to describe the evolution of the state variables $x_1(t)$, $x_{1,L}(t)$, $x_2(t)$, and $x_{2,L}(t)$, based on the model depicted in Figure 4. The model's dynamics is described by the following system of differential equations:

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} &= \lambda - x_1 + \mu\, x_{1,L}, & (6)\\ \dfrac{\mathrm{d}x_{1,L}}{\mathrm{d}t} &= (1-p)\, x_1 - \mu\, x_{1,L}, & (7)\\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} &= p\, x_1 - (\theta + \theta_1 + q_N^{x_2-1})\, x_2 + \theta_2\, x_{2,L}, & (8)\\ \dfrac{\mathrm{d}x_{2,L}}{\mathrm{d}t} &= \theta_1 x_2 - \theta_2 x_{2,L}. & (9) \end{cases}$$

with the constraints given below:

- $x_1$, $x_{1,L}$, $x_2$ and $x_{2,L}$ should be non negative,
- $\lambda_i > 0$, $\theta_1 > 0$, $\theta_2 > 0$, $\mu > 0$, and $\theta \geq 0$,
- $0 \leq p \leq 1$.

In what follows, we assume that $\theta = 0$, in order to simplify the model. Indeed, a system where devices often reach the maximum number of attempts is an unstable system, which we naturally try to avoid.

The model described in (5) is nonlinear and non-affine in the control. It can be easily demonstrated that the model described is unobservable, given its state $[x_1\ x_{1,L}\ x_2\ x_{2,L}]$ which cannot be precisely known. It is also uncontrollable because the blocking factor $p$ can only partially impact the state. These properties make the synthesis of an optimal controller guaranteeing the stability of the system, described above very complex.

Although the state is not observable, it is possible to produce an estimate of the average number of devices attempting access $\hat{x}_2$ by inverting Equations (1) and (4). This gives a very noisy measure, but may nevertheless be useful for blocking IoTs,

as demonstrated in [Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino*].

## IV. Reinforcement Learning-Based Access Controller for IoT Devices

The difficulty of observing the state of the system, described in the previous section, led us to consider strategies for inferring the blocking factor even in the presence of very noisy measurements. In this sense, we have focused on deep learning techniques, which have been very effective in automatically extracting system features in the presence of noisy or even incomplete data [Rolnick *et al.(2017)Rolnick, Veit, Belongie, and Shavit*].

Given the lack of data, we considered the Reinforcement Learning techniques class. More specifically, we considered the Twin Delayed Deep Deterministic policy gradient algorithm (TD3) technique, which can address a continuous action space, and which has been shown to be more effective in learning speed and performance than existing approaches [Fujimoto *et al.(2018)Fujimoto, van Hoof, and Meger*].

We formulate, in what follows, the problem of access in IoT as a reinforcement learning problem, in which an agent iteratively finds a suboptimal blocking factor, leading to a reduction of access contention.

### A. Problem Formulation

In reinforcement learning [Sutton and Barto(2018)], there are two main entities, an environment and an agent. The process of learning occurs through the interaction between these entities with the aim for the agent to optimize a total income. At each step $t$, the agent obtains a representation of the state $s_t$ of the environment and picks an action $a_t$, based on it. The agent thereafter applies this action on the environment. As a consequence, the environment passes into a new state $s_{t+1}$ and the agent receives a reward $r_t$ corresponding to this transition as well as the representation of the new state. This interaction can be modeled as a Markov Decision Process (MDP) $M = (S, A, P, R)$, with $S$ the state space, $A$ the action space, $P$ the transition dynamics, and $R$ the revenue. The behavior of the agent is defined by its policy $\pi : S \rightarrow A$, which allows a state to be associated with an action when it is a deterministic system, or a distribution of actions when it is stochastic. The objective of such a system is to find the optimal policy $\pi^*$ allowing for maximizing the cumulative revenue.

In the problem of IoT access control, we define an MDP, where the State, the Action, and the Revenue are defined as follows:

- *State:* Given the non-availability of the number of devices attempting the access at a given time $k$, the state we consider is based on the collected estimated values. Since a single measurement of this number is necessarily very noisy, we consider a series of several measurements, which we believe allow us to better reveal the current state of the network. The state $s_k$ is, thus, defined as the vector $(\hat{x}_2^k, \hat{x}_2^{k-1}, \ldots, \hat{x}_2^{k-H+1})$ where $H$ represents the measurement horizon.
  In our problem, $k$ progresses according to the preambles' arrival, whose frequency is constant.
- *Action:* At each step, the agent has to select the blocking factor $p$ that will be considered by the IoT objects. This value is continuous and deterministic, in the problem we are considering, i.e., the same state $s_k$ will always give the same action $a_k$.
- *Revenue:* This is a feedback signal received by the agent from the environment after the completion of an action. Thus, at step $k$, the agent obtains a revenue $r_k$ as a consequence of the action $a_k$ that was performed in the state $s_k$. This revenue will allow the agent to be informed of the quality of the executed action. The objective of the agent is to maximize this revenue.
  Note that the maximum number of successes is equal to the number of available preambles $N$, so this metric could be used as a parameter for the calculation of the revenue, which is given by the following equation:

$$r_k = \frac{1}{N\,H} \sum_{i=k-H+1}^{k} N_S^i \tag{10}$$

  where $N_S^i$ represents the number of successes at step $i$. Revenue is, therefore, maximized when the average revenue over a window of length $H$ is maximized. Unlike the measurement of the number of terminals attempting the access ($\hat{x}_2^k$), this measurement is not subject to noise, which allows a better quality of control. However, it should be noted that this is an indirect result of the number of attempts, making it more complex to handle.

The objective of such a system is to find the probability of blocking that maximizes the average reward. This is equivalent to reducing the distance between the real number of terminals attempting the access and the optimum. In order to achieve this objective, we rely on the TD3 algorithm.

The TD3 algorithm is an Actor–Critic approach, where the actor is a neural network that decides in a particular state of the action to be taken; the Critic network allows for knowing the value of being in a state and to choose a particular action. TD3 solves the problem of overvaluation in value estimation [Thrun and Schwartz(1993)], by introducing two Critic networks and taking the minimum between the two estimates. This approach is particularly interesting in our case given the inherent presence of measurement errors.

### B. Arrival Regulation System

The diagram in Figure 5 describes the system for controlling the number of attempts of IoT objects. This system is based on the diffusion of the blocking factor to the devices, through the System Information Blocks (SIBs) that are propagated, and, more specifically, through the SIB Type14 block, which allows for spreading the access blocking parameters [ETSI(2019)].

After receiving the blocking factor, the devices willing to make a transmission execute the ACB in order to proceed with the next steps, with a probability $p$, which is calculated

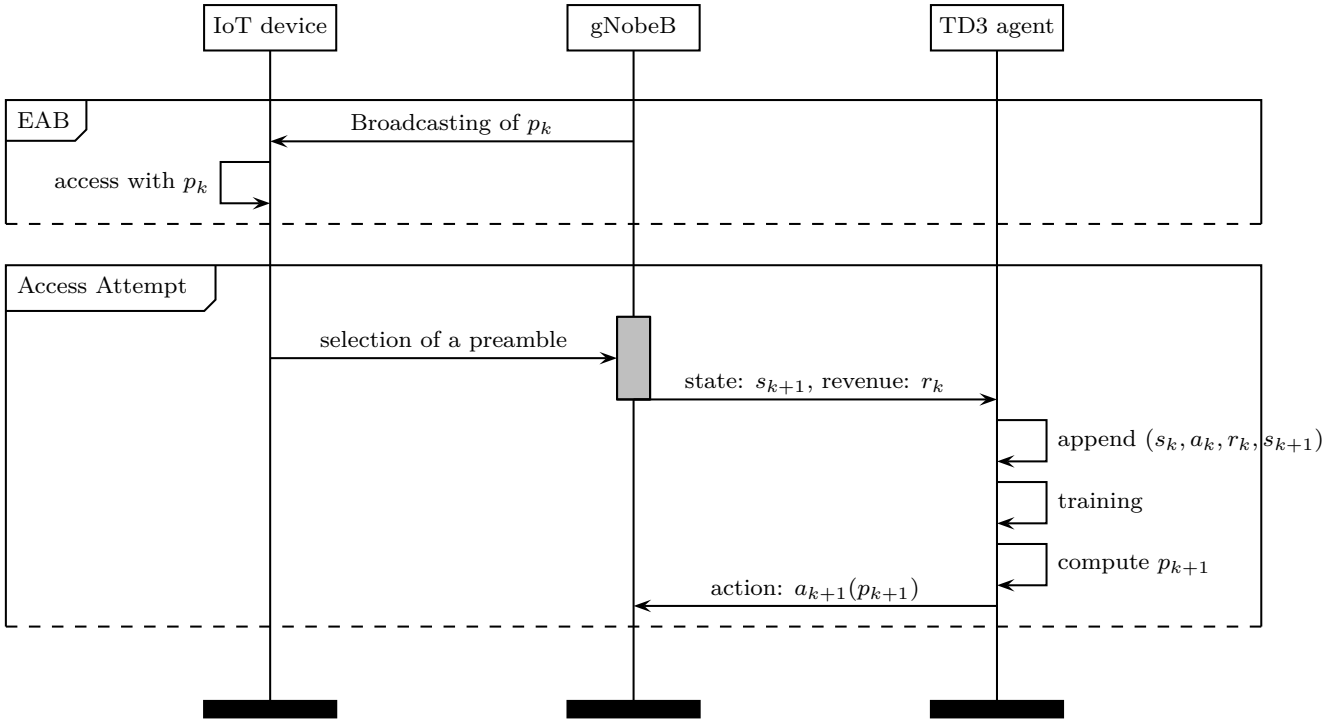



Fig. 5: Arrival regulation system.

by the proposed TD3-based controller. These devices can therefore attempt access by randomly selecting a preamble from the group of available preambles. Knowing the state of the preambles, the Base Station for 5G, also known as the gNodeB, can estimate the number of attempts that have been made. This estimate is very noisy, as the given model can only estimate averages in the case when there is at least one idle or successful preambles. To avoid too many variations in the estimate of the number of arrivals, we consider a moving average of this value.

The controller we have proposed receives these measures, together with the revenue, at the end of each access opportunity period. The revenue obtained will allow for determining the quality of the taken actions. These different elements are stored in a memory of past experiences. However, a random subset of this memory will allow for training the agent in a robust way, and choosing a new action.

These different actions are repeated periodically.

## V. Performance Evaluation

After having described our proposed access controller, we evaluate, in this section, its performance, using a discrete-event simulator built under Simpy[1].

We have considered an NB-IoT antenna in which the access requests arrive according to a Poisson's law with an average rate between two arrivals of 0.018 s. We considered a number of preambles $N$ equal to 12, with an arrival frequency equal to 0.1 s. In the considered system, each piece of equipment trying to access will be able to do it for a maximum of 16 times. Beyond this limit, the terminal abandons the transmission.

Unlike classical reinforcement learning problems, where the optimal value is generally not known, the optimum here is known and given by Equation (2). Thus, we compare the performance of our controller based on the TD3 technique to an adaptive approach, named ADAPT, and the Proportional Integral Derivative (PID) controller [Bouzouita *et al.(2015)Bouzouita, Hadjadj-Aoul, Zangar, Tabbane, and Viho*], which comes from control theory.

[1]https://simpy.readthedocs.io/en/latest/

The adaptive approach consists of gradually increasing the probability of blocking when the number of attempts is above a predefined threshold higher than the optimal value, in order to block more devices. When the value is below a predefined threshold under the optimal value, the probability of blocking is gradually reduced, in order to let more devices attempt the access.

We considered a measurement horizon $H$ equal to $10$ for both the TD3 and the PID controllers. The use of a larger measurement window does not allow a significant performance improvement, which means that a window of $10$ measurements sufficiently reflects the true state of the network.

Figure 6 shows the probability of blocking for the three considered strategies considered. The adaptive technique (see Figure 6a) starts with an access probability of 1 and adapts as traffic conditions change. The PID controller, in Figure 6b, has an access probability proportional to the estimation error, which makes the latter highly variable. For the strategy based on the TD3 algorithm, there is a first phase, lasting 1000 s, where the algorithm tries to explore the action space according to a uniform law (see Figure 6c). It is only after this phase that the algorithm begins to exploit its learning, which is refined as the experiments progress. One can note that, under TD3 (see Figure 6c), future actions are not related to past actions, contrary to the adaptive case. Indeed, the values of the actions can change completely, since they on-https://www.overleaf.com/project/601056afa11a1d217c80c8acly depend on the state of the network, which can change very quickly.

Figure 7 describes the impact of the control strategies described earlier on the average access latency. We do not consider in these plots the devices having failed to transmit, after a maximum number of attempts. We can see in Figure 7a,b that the latency using ADAPT and PID is generally of the same order. Although the latency using TD3 is slightly lower during the exploration phase, it increases during the exploitation phase to reach a latency of an order of magnitude comparable to other approaches. This means that the TD3 algorithm has no advantage in terms of latency.

Although TD3 does not have a particular advantage in terms of latency, it can be seen in Figure 8 that, after an exploration phase, the revenue improves very significantly. This reward is clearly superior to the ADAPT and the PID controllers. Indeed, in a steady state, the average reward in TD3 is around $29.25\%$ (see Figure 8c) while the reward for the ADAPT and the PID controllers is around $20.33\%$ and the $22.84\%$, respectively (see Figure 8a,b).

We can see in Figure 9a that different strategies have different results in terms of the number of successful preambles. The average number of successful preambles for the ADAPT, PID, and TD3 techniques are $2.47$, $2.74$, and $3.52$, respectively. Thus, the results obtained by ADAPT are the worst, followed by the PID controller, and finally the TD3-based strategy, which is clearly superior. These results represent a $42.51\%$ improvement over the ADAPT strategy and a $22.16\%$ improvement over the PID controller. The results obtained by TD3 are, thus, the closest to the optimum, which is equal to $4.61$ that is obtained from Equation (3).

The number of access successes directly reflects the quality of the strategy to control the number of devices attempting the access. From Equation (3), one can compute the optimal number of attempts which corresponds to an average of $11.49$ devices. With an average of $23.52$ accesses, ADAPT is the worst performing strategy. The results of the PID controller are equal to $17.15$ while the results of the TD3 strategy are equal to $15.70$. It can also be seen that the number of abandons remains relatively high in ADAPT with $4.48\%$ of the devices, while it is equal to $1.73\%$ with the PID controller, and less than $0.63\%$ for the TD3 strategy. This demonstrates the effectiveness of the proposed approach.

The results we have obtained in this paper show the superiority of the control through the reinforcement learning technique. This is due to several factors. The first factor concerns the variable being controlled. Indeed, the adaptive technique and the PID both use the control error, represented by the difference between the estimated value of the number of terminals accessing the network and the optimal number of terminals. The problem with this metric is the noise existing in the estimation of the number of terminals, as opposed to the number of successes, which is used in the revenue function, on which our solution is based. The second factor lies in the fact that deep learning techniques, in particular the TD3 algorithm on which our solution is based, allow us to better extract the true state of the network from noisy estimates, unlike the PID. The third factor resides in the fact that the learning technique, which we use, allows us to grasp a complex and highly nonlinear input pattern, which is not the case with the adaptive technique or even the PID.

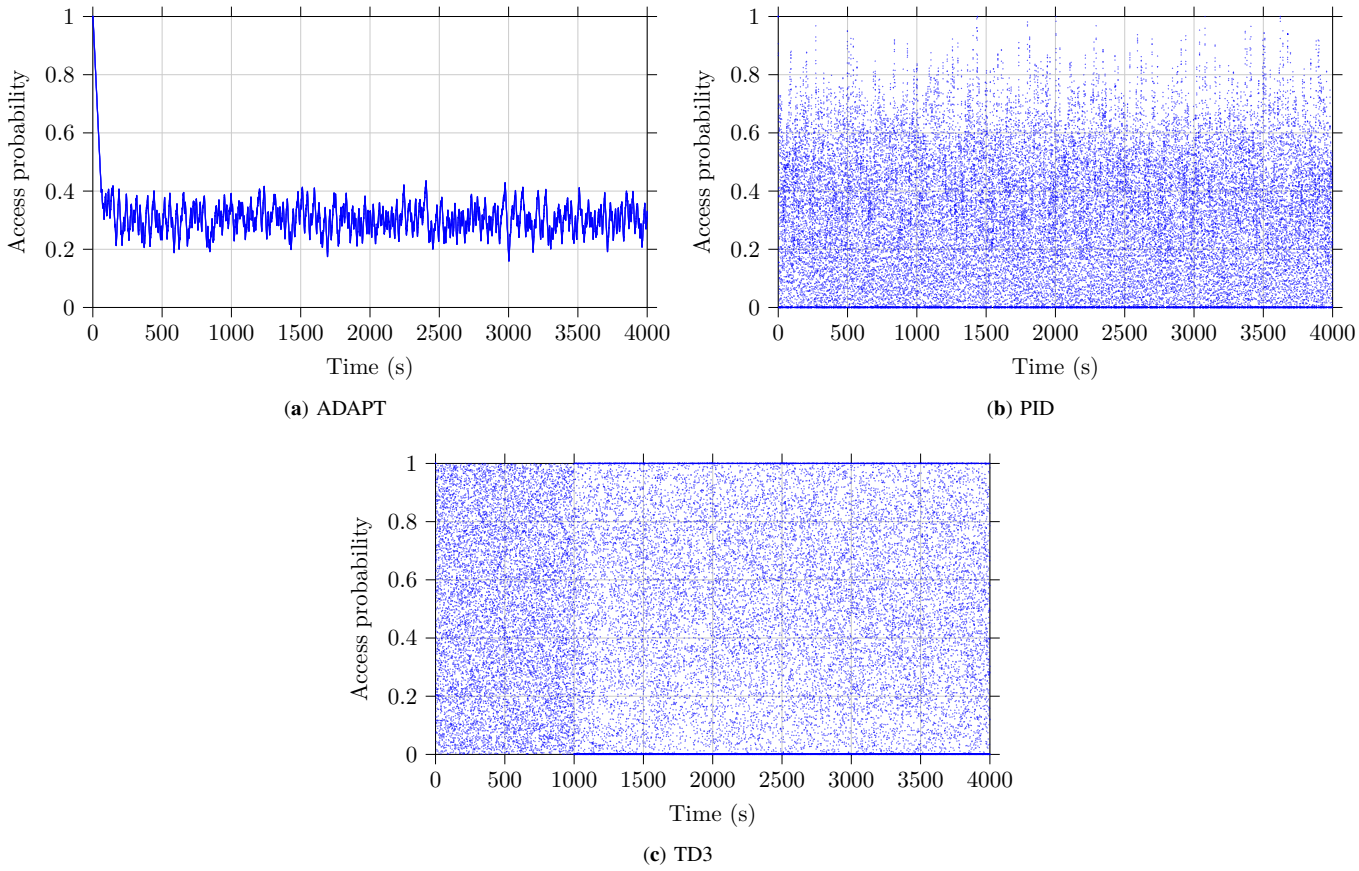
It should be noted that, by using the reinforcement learning approach, we can improve performance as we attempt the access. The limitation remains, however, the estimation errors that lead to errors in the state representation, hence the importance of having accurate estimators.

## VI. Conclusions

In this paper, we proposed a mechanism to control the congestion of the access network, which is considered as one of the most critical problems for IoT devices. We proposed to tackle the congestion at its root by effectively managing the random accesses of these devices through the use of the ACB mechanism.

The proposed access control mechanism is different from conventional methods, which are usually based on simple heuristics. Indeed, the proposed technique is based on recent advances in deep reinforcement learning, through the use of the TD3 algorithm. The proposed approach has, in addition, the advantage of learning from its environment and could therefore allow for adapting to the variation of the access pattern.

Simulation results show the superiority of the proposed approach, which manages to keep the number of access attempts close to the optimum, despite the lack of accurate




(**a**) ADAPT

(**b**) PID

(**c**) TD3

Fig. 6: The access probability for the considered strategies.

data on the number of access attempts. This work also shows the potential to use learning techniques in environments where the state cannot be known precisely.

In our future work, we plan to improve the estimation of the number of attempts using learning techniques.

## REFERENCES


[Lin *et al.(2016)Lin, Adhikary, and Eric Wang]* Lin, X.; Adhikary, A.; Eric Wang, Y. Random Access Preamble Design and Detection for 3GPP Narrowband IoT Systems. *IEEE Wirel. Commun. Lett.* **2016**, *5*, 640–643.

[Bouzouita *et al.(2016)Bouzouita, Hadjadj-Aoult, Zangar, and Tabbane]* Bouzouita, M.; Hadjadj-Aoult, Y.; Zangar, N.; Tabbane, S. On the risk of congestion collapse in heavily congested M2M networks. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 11–13 May 2016; pp. 1–5.

[3GPP(2011)] 3GPP. RAN Improvements for Machine-Type Communications; Technical Report (TR) 37.868, 3rd Generation Partnership Project (3GPP), Version 11.0.0.; 2011. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2630(accessed on 20 November 2020.)

[Bouzouita *et al.(2019)Bouzouita, Hadjadj-Aoul, Zangar, and Rubino]* Bouzouita, M.; Hadjadj-Aoul, Y.; Zangar, N.; Rubino, G. Estimating the number of contending IoT devices in 5G networks: Revealing the invisible. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3513, doi:10.1002/ett.3513.

[Fujimoto *et al.(2018)Fujimoto, van Hoof, and Meger]* Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1582–1591.

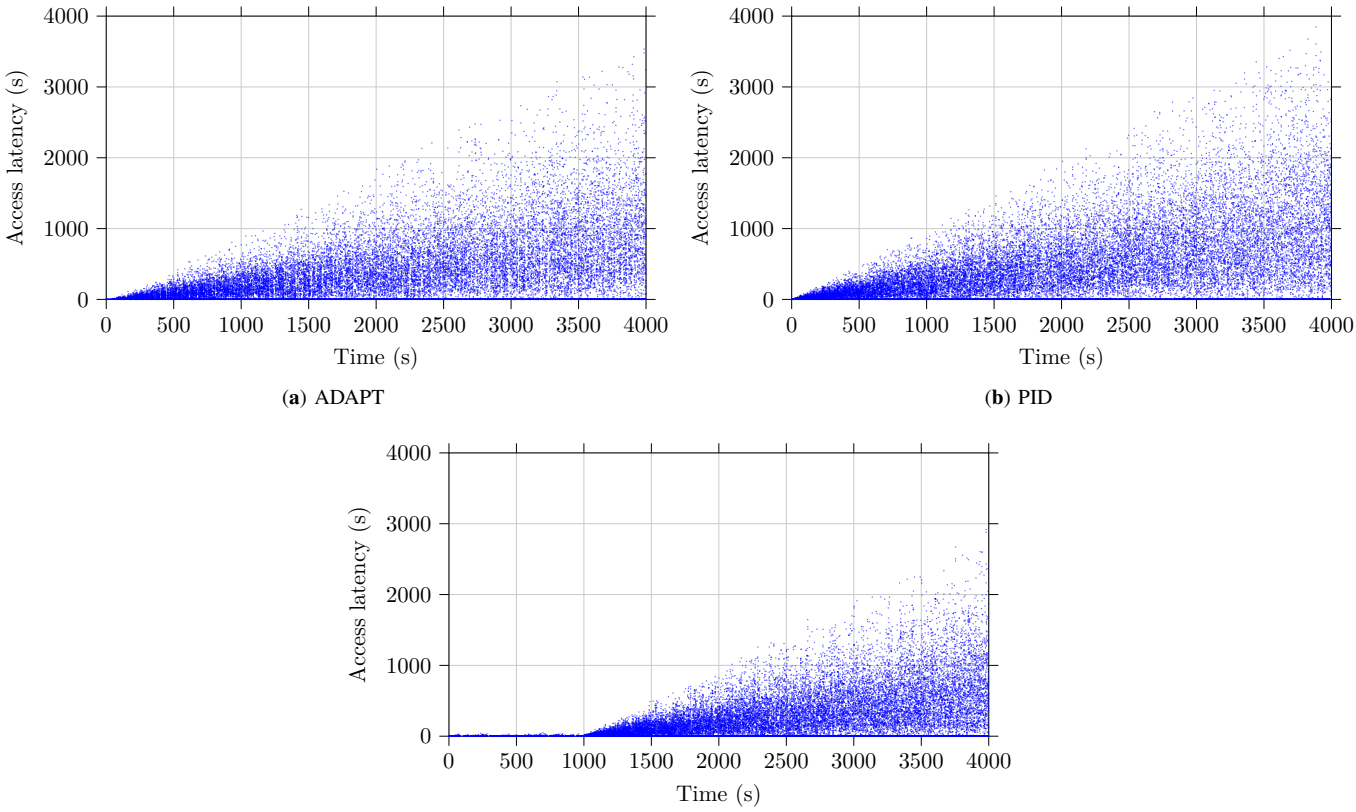
[3GPP(2015)] 3GPP. Cellular System Support For Ultra-Low Complexity and Low Throughput Internet of Things (CIoT); Technical Report (TR) 45.820, 3rd Generation Partnership Project (3GPP), Version 13.1.0.; 2015.Avaliable online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2719(accessedon20November2020).

[ETSI(2020a)] ETSI. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation; Protocol Specification (3GPP TS 36.211 version 16.3.0 Release 16), 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspxspecificationId=2425 (accessed on 23 October 2020).

[ETSI(2020b)] ETSI. Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol Specification; Protocol specification (3GPP TS 36.321 version 16.2.0 Release 16), 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2437 (accessed on 23 October 2020).

[ETSI(2019)] ETSI. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol Specification (3GPP TS 36.331 version 14.12.0 Release 14), 2019. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440 (accessed on 22 October 2020).

[Harwahyu *et al.(2019)Harwahyu, Cheng, Tsai, Hwang, and Bianchi]* Harwahyu, R.; Cheng, R.; Tsai, W.; Hwang, J.; Bianchi, G. Repetitions Versus Retransmissions: Tradeoff in Configuring NB-IoT Random




(**a**) ADAPT

(**b**) PID

(**c**) TD3

Fig. 7: The average latency of the devices for the considered strategies.


Access Channels. *IEEE Internet Things J.* **2019**, *6*, 3796–3805, doi:10.1109/JIOT.2019.2891366.

[Baracat and Brito(2018)] Baracat, G.H.; Brito, J.M.C. NB-IoT Random Access Procedure Analysis. In Proceedings of the 2018 IEEE 10th Latin-American Conference on Communications (LATINCOM), Guadalajara, Mexico, 14–16 November 2018; pp. 1–6, doi:10.1109/LATINCOM.2018.8613207.

[Jiang *et al.(2018)Jiang, Deng, Condoluci, Guo, Nallanathan, and Dohler]* Jiang, N.; Deng, Y.; Condoluci, M.; Guo, W.; Nallanathan, A.; Dohler, M. RACH Preamble Repetition in NB-IoT Network. *IEEE Commun. Lett.* **2018**, *22*, 1244–1247, doi:10.1109/LCOMM.2018.2793274.

[Harwahyu *et al.(2018)Harwahyu, Cheng, Wei, and Sari]* Harwahyu, R.; Cheng, R.; Wei, C.; Sari, R.F. Optimization of Random Access Channel in NB-IoT. *IEEE Internet Things J.* **2018**, *5*, 391–402, doi:10.1109/JIOT.2017.2786680.

[Sun *et al.(2018)Sun, Tong, Zhang, and He]* Sun, Y.; Tong, F.; Zhang, Z.; He, S. Throughput Modeling and Analysis of Random Access in Narrowband Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 1485–1493, doi:10.1109/JIOT.2017.2782318.

[Jeon *et al.(2018)Jeon, Seo, and Jeong]* Jeon, W.S.; Seo, S.B.; Jeong, D.G. Effective Frequency Hopping Pattern for ToA Estimation in NB-IoT Random Access. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10150–10154, doi:10.1109/TVT.2018.2857447.

[Hwang *et al.(2019)Hwang, Li, and Ma]* Hwang, J.; Li, C.; Ma, C. Efficient Detection and Synchronization of Superimposed NB-IoT NPRACH Preambles. *IEEE Internet Things J.* **2019**, *6*, 1173–1182, doi:10.1109/JIOT.2018.2867876.

[Zhang *et al.(5555)Zhang, Xie, and Wang]* Zhang, J.; Xie, D.; Wang, X. TARA: An Efficient Random Access Mechanism for NB-IoT by Exploiting TA Value Difference in Collided Preambles. *IEEE Trans. Mob. Comput.* **2020**, 1, doi:10.1109/TMC.2020.3019224.

[Ali *et al.(2017)Ali, Hossain, and Kim]* Ali, M.S.; Hossain, E.; Kim, D.I. LTE/LTE-A Random Access for Massive Machine-Type Communications in Smart Cities. *IEEE Commun. Mag.* **2017**, *55*, 76–83, doi:10.1109/MCOM.2017.1600215CM.

[Toor and Jin(2017)] Toor, W.T.; Jin, H. Comparative study of access class barring and extended access barring for machine type communications. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 18–20 October 2017; pp. 604–609, doi:10.1109/ICTC.2017.8191051.

[Park and Lim(2016)] Park, J.; Lim, Y. Adaptive Access Class Barring Method for Machine Generated Communications. *Mob. Inf. Syst.* **2016**, *2016*, 6923542:1–6923542:6, doi:10.1155/2016/6923542.

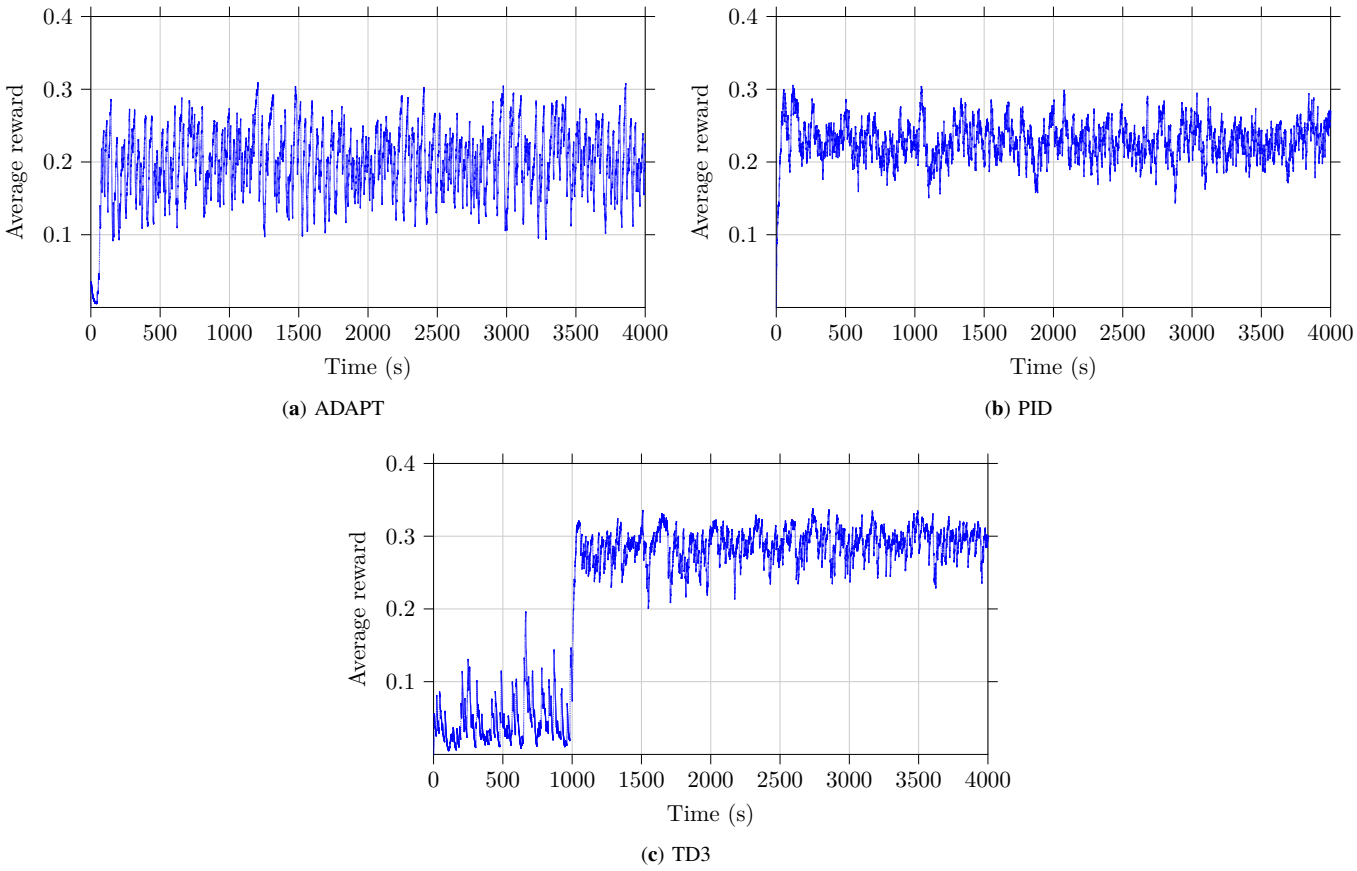
[Liu *et al.(2020)Liu, Agiwal, Qu, and Jin]* Liu, J.; Agiwal, M.; Qu, M.; Jin, H. Online Control of Preamble Groups with Priority in Massive IoT Networks. *IEEE J. Sel. Areas Commun.* **2020**, 1, doi:10.1109/JSAC.2020.3018964.

[Jin *et al.(2017)Jin, Toor, Jung, and Seo]* Jin, H.; Toor, W.T.; Jung, B.C.; Seo, J. Recursive Pseudo-Bayesian Access Class Barring for M2M Communications in LTE Systems. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8595–8599, doi:10.1109/TVT.2017.2681206.

[Cheng *et al.(2015)Cheng, Chen, Chen, and Wei]* Cheng, R.; Chen, J.; Chen, D.; Wei, C. Modeling and Analysis of an Extended Access Barring Algorithm for Machine-Type Communications in LTE-A Networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 2956–2968, doi:10.1109/TWC.2015.2398858.

[Agiwal *et al.(2019)Agiwal, Maheshwari, and Jin]* Agiwal, M.; Maheshwari, M.K.; Jin, H. Power Efficient Random Access for Massive NB-IoT Connectivity. *Sensors* **2019**, *19*, 4944, doi:10.3390/s19224944.

[Bouzouita *et al.(2015)Bouzouita, Hadjadj-Aoul, Zangar, Rubino, and Tabbane]* Bouzouita, M.; Hadjadj-Aoul, Y.; Zangar, N.; Rubino, G.; Tabbane, S. Multiple Access Class Barring Factors Algorithm for M2M Communications in LTE-Advanced Networks. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '15), Cancun, Mexico, 2–6 November 2015; Association for




(**a**) ADAPT

(**b**) PID

(**c**) TD3

Fig. 8: The average reward of the considered strategies.

Computing Machinery: New York, NY, USA, 2015; pp. 195–199, doi:10.1145/2811587.2811624.

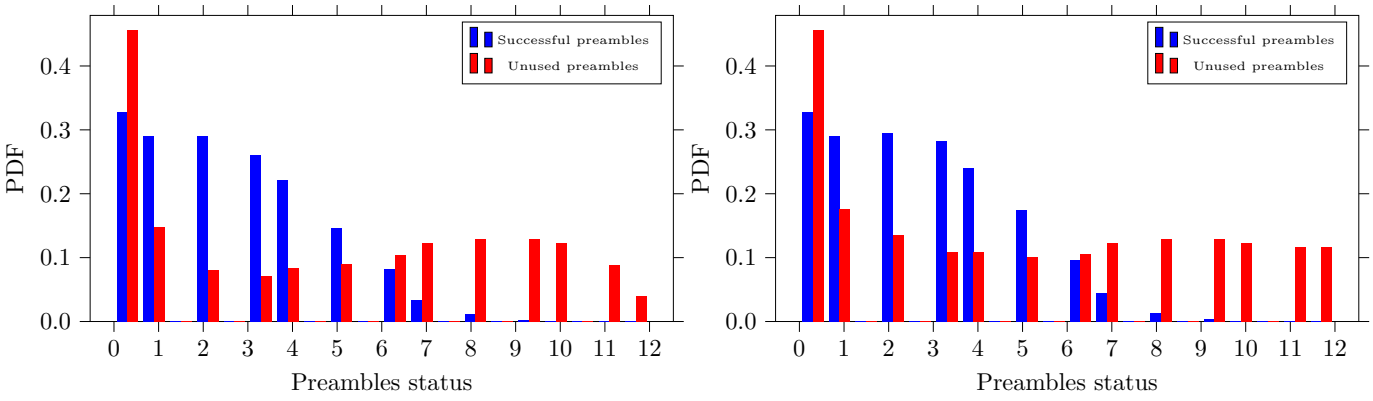
[Rolnick *et al.(2017)Rolnick, Veit, Belongie, and Shavit]* Rolnick, D.; Veit, A.; Belongie, S.J.; Shavit, N. Deep Learning is Robust to Massive Label Noise. *arXiv* **2017**, arXiv:1705.10694.

[Fujimoto *et al.(2018)Fujimoto, van Hoof, and Meger]* Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv* **2018**, arXiv:1802.09477.

[Sutton and Barto(2018)] Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; A Bradford Book: Cambridge, MA, USA, 2018. .

[Thrun and Schwartz(1993)] Thrun, S.; Schwartz, A. Issues in Using Function Approximation for Reinforcement Learning, 1993. Avaliable online: https://www.ri.cmu.edu/pub_files/pub1/thrun_sebastian_1993_1/thrun_sebastian_1993_1.pdf (accessed on 20 November 2020).

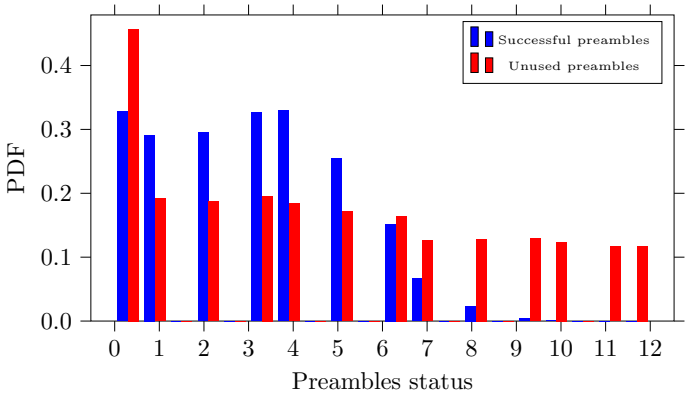
[Bouzouita *et al.(2015)Bouzouita, Hadjadj-Aoul, Zangar, Tabbane, and Viho]* Bouzouita, M.; Hadjadj-Aoul, Y.; Zangar, N.; Tabbane, S.; Viho, C. Chapter 20—A random access model for M2M communications in LTE-advanced mobile networks. In *Modeling and Simulation of Computer Networks and Systems*; Obaidat, M.S., Nicopolitidis, P., Zarai, F., Eds.; Morgan Kaufmann: Boston, MA, USA, 2015; pp. 577–599, doi:10.1016/B978-0-12-800887-4.00020-1.




(**a**) ADAPT




(**b**) PID




(**c**) TD3

Fig. 9: The status of the preambles with the different approaches.