Random Access Control in NB-IoT with Model-Based Reinforcement Learning

Juan J. Alcaraz, Juan-Carlos Sanchez-Aarnoutse, Alejandro-Santos, Martinez-Sala, Francisco-Javier Gonzalez-Castaño,

***Abstract*—In NB-IoT, the cell can be divided into up to three coverage extension (CE) levels, each associated with a narrowband Physical Random Access Channel (NPRACH) that has a CE level-specific configuration. Providing resources to NPRACHs increases the success rate of the random access procedure but detracts resources from the uplink carrier for other transmissions. To effectively address this trade-off we propose to adjust the NPRACH parameters along with the power thresholds that determine the CE levels, which allows to control at the same time the traffic distribution between CE levels and the resources allocated to each CE level. Since the traffic is dynamic and random, reinforcement learning (RL) is a suitable approach for finding an optimal control policy, but its inherent sample inefficiency is a drawback for online learning in an operational network. To overcome this issue, we propose a new model-based RL algorithm that achieves high efficiency even in the early stages of learning.**

***Index Terms*—Narrowband Internet of things (NB-IoT), reinforcement learning, NPRACH, resource allocation.**

I. Introduction

NARROWBAND Internet of Things (NB-IoT) is an IoT cellular technology specified by the Third Generation Partnership Project (3GPP) to provide efficient connectivity to a massive number of low-complexity devices, and is crucial for machine-type communications in 5G and beyond networks [1]. It finds applications in fields like smart metering, logistics, tracking, and smart cities. Its radio interface operates on a minimal bandwidth of 180 kHz for both downlink and uplink, facilitating deployment within legacy LTE networks or independently, for instance, using a GSM carrier [2]. Its physical layer [3]–[5] utilizes repetition and signal combining techniques to extend reach to low power devices in unfavorable locations [6]. NB-IoT introduces three coverage enhancement (CE) levels to accommodate devices under diverse path loss conditions, ensuring equitable access. Each CE level has specific time-frequency resources in the uplink carrier, known as narrowband Physical Random Access Channel (NPRACH), which provide access opportunities to the devices of each CE level.

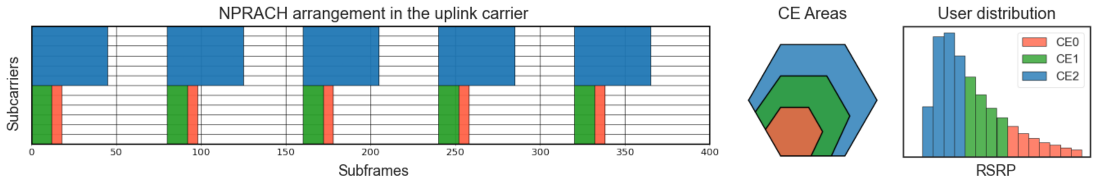
To access the network, a device must first assess its CE level by measuring its reference signal receive power (RSRP), and compare it to the RSRP thresholds of each CE level. These thresholds are configured and broadcast by the base station periodically. The device must then initiate a random access (RA) procedure, similar to a framed slotted ALOHA, in which the device transmits a signal (preamble) over the NPRACH of its CE level. To reduce collisions, each device can choose randomly among the set of orthogonal preambles available in the NPRACH, determined by the amount of 3.75 KHz subcarriers assigned to that NPRACH. For each CE level, the preambles are repeated a predefined number of times to ensure a high probability of detection by the base station. The NPRACH appears periodically in the uplink carrier based on a predefined period.

The configuration of the number of subcarriers and the period length of the NPRACH raises a challenging trade-off, since allocating more subcarriers and shorter periods provides more access opportunities to devices but detracts more resources from the uplink carrier, which are needed for other transmissions. Therefore, the optimal setting of NPRACH resources depends on the traffic load generated from each CE level, (*i.e.* number of devices, access attempt rate, and generated data), while the traffic load of each CE level depends on the RSRP threshold configuration. Although there have been previous works [7]–[9] studying the configuration of NPRACH parameters, none have addressed it together with the definition of the CE levels. Figure 1, illustrates how intertwined these two mechanisms are. The RSRP thresholds determine the range of each CE area, and thus the distribution of users by CE levels. It also determines the number of preamble repetitions in each NPRACH, which impacts their resource requirements. NPRACHs can be arranged according to multiple combinations of periods and subcarriers. The 4 configurations shown in Figure 1 give an example of the variability of the NPRACH arrangements.

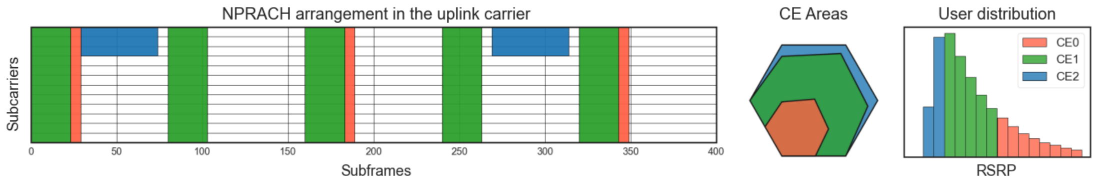
Given the characteristics of the radio recievers at the base station, the number of preamble repetitions can be predetermined for each RSRP level. However, the remaining NPRACH parameters depend on the spatial distribution of the incomming traffic, which is time-varying, random, and unknown *a priori*. To address this challenge, we propose an adaptive mechanism that adjusts these parameters dynamically based on the observations obtained from the network in operation. This type of problem fits perfectly in the reinforcement learning

The work of J.J. Alcaraz, J.-C. Sanchez-Aarnoutse and A.-S., Martinez-Sala was supported by Grant PID2020-116329GB-C22 funded by MICIU/AEI/10.13039/501100011033. The work of F.-J. Gonzalez-Castaño was supported by Grant ED481B-2022-019 funded by Xunta de Galicia (Spain), and by Grant PID2020-116329GB-C21 funded by MICIU/AEI/10.13039/501100011033.

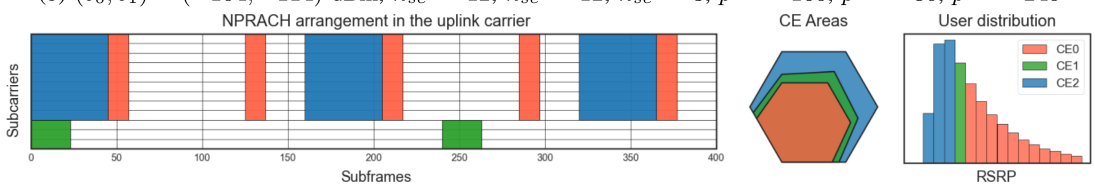
J.J. Alcaraz, J.-C. Sanchez-Aarnoutse and A.-S., Martinez-Sala are with the Department of Information and Communication Technologies of the Technical Universitiy of Cartagena (UPCT), Spain (e-mail: juan.alcaraz@upct.es; juanc.sanchez@upct.es; AlejandroS.Martinez@upct.es.) F.-J. Gonzalez-Castaño is with the Telematics Engineering Department, University of Vigo, Spain. (e-mail: javier@det.uvigo.es)



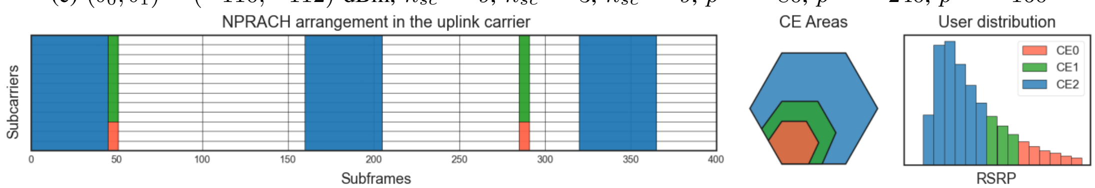
(a) $(\delta_0, \delta_1) = (-102, -110)$ dBm, $n_{sc}^{(0)} = 6$, $n_{sc}^{(1)} = 6$, $n_{sc}^{(2)} = 6$, $p^{(0)} = 80$, $p^{(1)} = 80$, $p^{(2)} = 80$



(b) $(\delta_0, \delta_1) = (-104, -114)$ dBm, $n_{sc}^{(0)} = 12$, $n_{sc}^{(1)} = 12$, $n_{sc}^{(2)} = 3$, $p^{(0)} = 160$, $p^{(1)} = 80$, $p^{(2)} = 240$



(c) $(\delta_0, \delta_1) = (-110, -112)$ dBm, $n_{sc}^{(0)} = 9$, $n_{sc}^{(1)} = 3$, $n_{sc}^{(2)} = 9$, $p^{(0)} = 80$, $p^{(1)} = 240$, $p^{(2)} = 160$



(d) $(\delta_0, \delta_1) = (-100, -106)$ dBm, $n_{sc}^{(0)} = 3$, $n_{sc}^{(1)} = 9$, $n_{sc}^{(2)} = 12$, $p^{(0)} = 240$, $p^{(1)} = 240$, $p^{(2)} = 160$

Fig. 1: Diverse arrangements of the NPRACH resources in the uplink carrier, CE areas, and device distribution over RSRP values, for different configurations of the RSRP thresholds (δ_0, δ_1) in dBm, and the NPRACH parameters. These parameters include the number of subcarriers $n_{sc}^{(i)}$ and the periodicity $p^{(i)}$ (in subframes), for CE levels $i = 0, 1, 2$. The cell corresponds to the coverage area of a sector antenna. The power levels and the range of the CE areas in the cell have been generated with the simulation environment described in Section IV.

(RL) domain. However, conventional RL algorithms, even the most advanced ones, are very sample inefficient, which, together with the need to explore the action space extensively, result in poor network performance during the initial learning stages, which can last from a few hours to several days. For this reason, we propose a new model-based RL approach that drastically improves sample efficiency and the initial performance.

A. Related Works and Contributions

In recent years, NB-IoT has attracted significant research interest, with works focusing on the configuration and optimization of diverse system parameters and control mechanisms. This includes, for example, the allocation of time-frequency resources to narrowband Physical Uplink Shared Shannel (NPUSCH) transmissions, [10]–[13], the selection (scheduling) of devices for downlink [14] or uplink [15] transmissions, link-adaptation of NPUSCH transmissions [16], [17], uplink scheduling in combination with link-adaptation [18], and the configuration of the downlink signalling channel NPDCCH [19], [20].

In this context, the analysis and configuration of the RA procedure stands out as a particularly relevant area of study. There are several mechanisms involved in RA that need to be carefully fine-tuned to to ensure the efficiency and stability of the network, like access class barring (ACB) or backoff (BO) schemes, aimed at preventing congestion when too many devices attempt to access at the same time, and configured in [21], [22] by means of analytic models. The analytic approach was also used to compute the success probability of the RA procedure in [23], [24], and to estimate the number of devices attempting to access the network (traffic load) [25], [26].

A crucial aspect of RA is the configuration of NPRACH resources because of the trade-off between increasing the RA success rate and ensuring sufficient resources for NPUSCH transmissions. This trade-off was studied in [7] and addressed by a means of an analytical model in [8]. Other aspects of NPRACH configuration were considered in [27], focused on the optimization of preamble repetitions, and [28], which proposed a heuristic approach for the adjustment of several RA parameters including the number of NPRACH subcarriers and the BO timer. The closest antecedent to our work is [9], which proposes the use of RL to determine the NPRACH subcarriers and periodicity for each CE level. Our work is fundamentally different from this one for two reasons: first, because we extend the scope of the problem by also considering the configuration of RSRP thresholds, and second, because we propose a novel model-based approach to overcome the inherent limitations of conventional RL algorithms in this environment.

There are other precedents of the use of RL for controling NB-IoT functionalities. For example, [29] employs a deep RL algorithm (DDPG) for the configuration of ACB, and [30] proposes the use of multiple agents for controlling the ACB, BO, and *distributed queueing* (DQ) mechanisms in parallel. RL has been used in [18] for NPUSCH scheduling and link-adaptation, and has been also used for random access in heterogeneous networks in [31], and for dynamic multi-channel access, in [32]. As pointed out in [9], RL algorithms need, in general, to be trained offline in a simulator before deployment and, once deployed, real network conditions can be very different from the simulation environment, rendering control policies ineffective.

Our proposal is especially tailored to learn autonomously once deployed in the network (*online learning*), a task that state-of-the-art RL algorithms struggle to accomplish without deteriorating the transmission delay during their initial stages of learning, as verified by our numerical results in Section IV. The reason is the low sample efficiency of model-free RL (MFRL) algorithms. They need to explore multiple policies before converging to an efficient one, which implies selecting very ineffective actions during long periods. To overcome this limitation, we leverage the higher sample efficiency of model-based RL (MBRL) [33], [34] with respect to MFRL.

This work is framed within a broader trend focused on developing online learning algorithms to control network functions. Previous works in this line have addressed resource allocation for network slices [35], interference coordination in LTE [36], [37], energy saving for small cells, [38], and uplink transmission control in NB-IoT [18], using diverse approaches based on multi-armed bandits [36], sequential likelihood ratio tests [37], bayesian models [38], kernel-based methods [35], and multi-agent MBRL [18]. This is the first work focused on NPRACH configuration together with the definition of CE levels in NB-IoT. Our contributions are:

- We present and address problem of RA control in NB-IoT, which is more general than previous ones as we aim at maximizing the resource efficiency by controlling not only the resource allocated to each NPRACH but also the incoming traffic at each NPRACH.
- We propose a new MBRL approach whose high sample efficiency facilitates online learning. Specifically, our scheme prevents the low performance associated to the action space exploration in the early stages of a learning episode.
- We develop a novel agent architecture that combines modeling techniques historically used for network design (queueing theory, combinatorial analysis, maximum likelihood estimation), and the learning and control capabilities of RL, leveraging both approaches.

The rest of the paper is organized as follows: Section II describes the NB-IoT mechanisms related to random access and uplink transmission, and formulates the problem within the RL framework. Section III details our novel model-based proposal. Section IV explains our evaluation methodology, describes the baselines with which we compare our proposal, and presents the numerical results. Finally, Section V summarizes our findings and points out future enhancements to our proposal.

II. System Description and Problem Statement

A. System Description

The system under study consists of a base station serving several thousands of NB-IoT devices. These devices must complete a random access procedure, making use of the NPRACH resources, a connection procedure and, once connected, they must wait for the base station to send them an uplink grant for a NPUSCH data tranmission. We consider one carrier for each transmission direction (uplink and downlink) with a bandwidth of 180 KHz and the time dimension divided into frames comprising 10 subframes of 1 ms.

1) Coverage enhancement levels: The coverage area of the eNB is divided into several zones called coverage enhancement levels (CE levels) to address the different radio conditions. Up to three CE levels can be defined by the eNB through power thresholds according to the requirements of the network. These thresholds (δ_0 and δ_1) are based on the values of the RSRP (reference signal receive power). The RSRP value is computed at each UE by averaging the power received over a specific set of resource elements in the downlink carrier. The UE compares its measured RSRP with the threshold levels to determine which CE level it belongs to:

- If RSRP $\leq \delta_1$, the UE belongs to CE level 2 (CE2)
- If $\delta_1 <$ RSRP $\leq \delta_0$, the UE belongs to CE level 1 (CE1)
- If $\delta_0 <$ RSRP, the UE belongs to CE level 0 (CE0)

2) Random access through the NPRACH: The NB-IoT carrier contains an NPRACH for each CE level. To request access to the network, a UE has to initiate a RA procedure by transmitting a particular signal through the NPRACH corresponding to their CE level. This signal, known as *preamble*, consists of four groups of symbols, and each group forms a pure sinusoid whose frequency is determinied according to a predefined frequency hopping sequence. The preamble sequence must be repeated a number of times in order to guarantee a high detection probability. The number of repetitions depends on the CE level and is specified by a NPRACH parameter $n_{rep}^{(i)}$, for $i = 0, 1, 2$, that can be configured to 1, 2, 4, 8, 16, 32, 64, or 128.

The subcarrier spacing used in NPRACH is 3.75 kHz, leading to 48 possible frequencies (subcarriers) over the 180 kHz bandwidth of the NB-IoT system. An NPRACH can be configured to use 12, 24, 36 or 48 contiguous subcarriers. We use $n_{sc}^{(i)}$ to denote the number of subcarriers of the NPRACH of CE level i for $i = 0, 1, 2$. The initial frequency selected by the UE, known as RAPID (random access preamble identifier), is associated to one of the predefined frequency hopping sequences, such that devices selecting a different RAPID do not collide. The larger $n_{sc}^{(i)}$, the more deterministic hopping sequences can be used by the UEs of CE level i, hence reducing the probability of collisions in this CE level. The NPRACH of each CE appears periodically in the carrier with a period that can be configured to 40, 80, 160, 320, 640, 1280, or 2560 ms. We use $p^{(i)}$ to denote the period of the NPRACH of CE level i for $i = 0, 1, 2$. For each CE level, $p^{(i)}$ and $n_{sc}^{(i)}$ should be determined according to the data traffic profile of the devices in this CE level, while $n_{rep}^{(i)}$ should be configured according to the worst-case pathloss in the CE level.

3) Connection establishment: After detecting the preambles, the eNB sends a RAR (random access response) message, called *msg2*, to the devices whose preambles have been detected. The UE devices that have transmitted a preamble, expect to receive this *msg2* within a RAR window, otherwise they will start a new RA procedure. The *msg2* includes a C-RNTI (cell radio network temporary identifier) that is unique to each preamble, and contains an uplink grant allowing the UE to transmit its response, called *msg3*. The uplink carrier bandwidth not reserved for NPRACH is divided into 12 subcarriers of 15 kHz. Thus the uplink grant allocates a specific set of subframes and subcarriers in the uplink carrier for the *msg3* transmission, avoiding any overlap with other transmissions, and specifies the link-adaptation parameters (modulation and coding scheme, and repetitions) for the transmission. If the *msg3* is received by the eNB, it is responded with a *msg4* message that ends the access procedure and sets the device to connected state. If the eNB cannot decode a *msg3*, it can send an uplink grant for *msg3* retransmission using an HARQ scheme.

When the base station manages to decode a collided preamble, several stations will receive the same *msg2*, resulting in several colliding *msg3* responses. Note that the eNB can recognize eventual preamble collisions occurred in an NPRACH as corrupted *msg3* responses [7], [39]. When a device sends a *msg3*, it starts a contention timer. If this timer expires before the reception of the *msg4* response, the device will start a new access attempt. The eNB can specify a maximum backoff period for UEs that failed to set up a connection.

4) Data transmission over the NPUSCH: Once connected, a UE waits for an uplink grant from the eNB allowing a NPUSCH data transmission in the uplink carrier. The UE will transmit a transport block over the allocated resources with the prescribed link-adaptation configuration. Then, the device can receive either an ACK, if the packet was decoded, or another uplink grant for an HARQ retransmission, if the packet was not decoded. After receiving an ACK for the transmitted packet, the device disconnects from the base station.

5) Signaling channels: The NPBCH (narrowband Physical Broadcast Channel) is the first to be decoded by the UEs before initiating an access procedure. The NPBCH carriers essential information like the system information block 2 (SIB2-NB) which holds the configuration of the RSRP thresholds and the NPRACH resources. The SIB2-NB is regularly transmitted based on a predefined update period.

The base station uses the NPDCCH (narrowband Physical Downlink Control Channel) to transmit control messages to the UEs that have initiated an RA procedure or are already connected to the network. NPDCCHs appear periodically on the downlink carrier. The number of subframes and the period of the NPDCCH are system parameters. Control information, such as the *msg2* or *msg4* messages, is carried in a logical block called downlink control information (DCI). DCIs are repeated a number of times depending on the pathloss or CE level of the destination device.

B. Problem description

We consider a base station that automatically configures the parameters that regulate random access in NB-IoT: the RSRP thresholds (δ_0, δ_1), and the NPRACH parameters $n_{sc}^{(i)}$, and $p^{(i)}$ for $i = 0, 1, 2$. This information is updated and broadcast to the devices in the periodically transmitted SIB2-NBs. Therefore, between consecutive updates, the base station must observe the evolution of relevant system variables (e.g. number of detected preambles, number of collisions, etc.) and select the most suitable configuration for the upcoming update, in order to maximize the number of devices that complete their NPUSCH transmission. The station must learn the control policy autonomously by interacting with the network.

When configuring the NPRACH parameters we face a fundamental trade-off: Assigning more resources to NPRACH channels increases the preamble detection rate, but reduces the available resources for *msg3* and NPUSCH transmissions. Choking the *msg3* transmission rate limits the UE throughput, and insufficient resources for NPUSCH lead to an unbounded increase in delay, and even to stuck NPUSCH transmissions when it is not possible to fit them in the carrier.

The preamble repetitions $n_{rep}^{(i)}$ in the NPRACH of each CE level i, can be pre-configured based on offline experimental measurements, *i.e.* for a given pair of threshold values $\delta = (\delta_0, \delta_1)$, the value of $n_{rep}^{(i)}$ is determined by the largest loss in CE level i and the desired preamble detection probability (see Section IV for further details). The link-adaptation parameters for *msg3* transmissions in each CE level can be determined similarly.

Therefore, we see that the selection of δ is highly intertwined with the NPRACH configuration: first, it determines the incoming traffic load for each NPRACH, and second, it affects the amount of time-frequency resources consumed by NPRACHs and *msg3* messages. To properly configure δ, the control agent should consider how the devices are distributed among the possible RSRP levels.

Summarizing, the control problem addressed consists of determining, for each SIB2-NB update period, the RSRP thresholds (δ_0, δ_1) and the NPRACH parameters, $n_{sc}^{(i)}$ and $p^{(i)}$ for $i = 0, 1, 2$, maximizing the UE throughput in terms of data messages sent (successful NPUSCH transmissions) per second. Both the incoming traffic intensity and the distribution of users are random and unknown *a priori* to the base station; therefore, the configuration policy of these parameters must be learned online by the control algorithm.

In this work we explore two approaches for addressing this problem: one based on model-free (MF) RL and the other based on model-based (MB) RL.

Variable	Meaning
$n_{sc}^{(i)}$	number of subcarriers allocated to NPRACH for CE level i
$p^{(i)}$	NPRACH periodicity in ms for CE level i
$c^{(i)}$	NPRACH configuration for CE level i: $c^{(i)} = (n_{sc}^{(i)}, p^{(i)})$
δ_0, δ_1	RSRP threshold levels for CE levels 0 and 1 respectively
δ	a specific RSRP threshold configuration $\delta = (\delta_0, \delta_1)$
Δ	set of RSRP threshold configurations
$l_1, \ldots, l_L$	Possible RSRP thresholds levels
$h(l_k)$	estimated fraction of devices whose RSRPs lie within the range (l_{k-1}, l_k)
$h^{(i)}(\delta)$	fraction of UEs belonging to CE level i given δ
$e^{(i)}(\delta)$	*msg3* reception probability for CE level i given δ
$\lambda^{(i)}$	arrival rate for CE level i UEs
$\widehat{\lambda}$	overall estimated arrival rate
β	adjustment factor for the estimated arrival rate
$\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)$	maximum NPRACH detection rate in CE level iÂ·
$n_{sf}^{(i)}(\delta)$	number of subframes of the NPRACH for CE level i
$r^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)$	ratio of resources consumed by the RA of CE level i
$P_n(d, c, n_{sc})$	Probabilty of observing d non-colliding preambles when n UEs chose among n_{sc} preambles
$A^\star(d, c, n_{sc})$	lookup table mapping (d, c, n_{sc}) tuples to estimated UE access attempts in a NPRACH
$m_{\max}^{(i)}$	maximum *msg3* transmissions for CE level i per NPRACH period
p_{DCCH}	NPDCCH period in subframes
w_{RAR}	*ResponseWindowSize* parameter in NPDCCH periods
$n_{RU}^{(i)}$	number of RUs of a *msg3* for CE level i
$n_{DCI}^{(i)}$	number of DCI repetitions for CE level i
$w_{RU}^{(i)}$	RUs per RAR window for CE level i
$w_{DCI}^{(i)}$	DCIs per RAR window for CE level i
$D^\star(i, n_{sc}^{(i)}, p^{(i)}, \delta)$	lookup table mapping $(i, n_{sc}^{(i)}, p^{(i)}, \delta)$ tuples to the expected number of UEs detected per NPRACH in CE i
$c_i^\star(\delta, \beta\widehat{\lambda})$	best possible NPRACH configuration for CE level i at a given δ and $\beta\widehat{\lambda}$
$(\beta\widehat{\lambda})_i$	elements of a finite set of evenly spaced values of $\beta\widehat{\lambda}$
$C^\star$	lookup table mapping $(\beta\widehat{\lambda})_i$ values to the optimal configuration of δ and NPRACH parameters

TABLE I: Definitions of variables used in the paper.

C. Reinforcement Learning Elements

In this subsection we present the elements common to both RL approaches: the objective function, the observation of the system, and the selected action.

Reinforcement Learning (RL) is a type of machine learning algorithm where an agent learns to make decisions by interacting with the environment. The agent's goal is to maximize the expected value of a cumulative reward, which is the sum of rewards received over time. At each time step t, the agent observes the current state S_t of the environment and selects a control action A_t which is applied to the environment. As a result of this action, the agent receives a reward R_{t+1} and the environment transitions to a new state S_{t+1}. The agent's objective is to learn a policy π, which is a mapping from states to actions, such that the expected value of the sum of discounted rewards (return) is maximized. The discount factor $\gamma \in [0, 1]$ controls the importance of future rewards. The expected value of the return from t is given by:

$$\mathbb{E}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+i}\right] \quad (1)$$

where the expectation is taken over the probabilities of the trajectories (sequences of state-action pairs) determined by π. In our environment, the **time steps** $t = 0, 1, 2, \ldots$ are the instants when the agent makes a configuration decision (*i.e.* the SIB2 updates).

Action: A_t contains the RSRP thresholds for CE0 and CE1 (δ_0, and δ_1) and the NPRACH configuration parameters for CE0 ($n_{sc}^{(0)}$, $p^{(0)}$), CE1 ($n_{sc}^{(1)}$, $p^{(1)}$), and CE2 ($n_{sc}^{(2)}$, $p^{(2)}$).

Observation: S_t contains a selection of the observable variables of the system that can be useful for an agent to configure the RSRP levels and the NPRACHs:

- For each CE, average preamble detections and average preamble collisions per NPRACH.
- For each CE, the ratio between received *msg3* messages and preamble detections.
- The ratio of carrier resources devoted to RA.
- The ratio of non-RA carrier resources occupied by NPUSCH transmissions.
- Total number of connected UEs.
- Estimation of the RSRP distribution, h.

To simplify notation, let $h(l_k)$ for $k = 1, \ldots L+1$ denote the ratio of UEs whose reported RSRPs lie within the interval $(l_{k-1}, l_k]$, where l_0 and l_{L+1} are two auxiliary levels corresponding to $-\infty$ and $+\infty$ respectively. The update of h can be efficiently performed with an incremental update, as follows:

$$h(l_k) = k(l_k) + \frac{1}{n}(\mathbb{I}_{\{\text{RSRP}_n \in (l_{k-1}, l_k]\}} - h(l_k)) \quad (2)$$

for $k = 1, \ldots, L+1$, where n is a counter of connected UEs, and $\mathbb{I}_{\{\text{RSRP}_n \in (l_{k-1}, l_k]\}}$ is an indicator function that equals 1 if the RSRP of the n-th UE lies within $(l_{k-1}, l_k]$, and equals 0 otherwise.

Reward: R_t simply indicates the number of NPUSCH successfully transmitted during the last SIB2 update period.

III. Model-Based Approach

A. Design Principles of the Model

Our model divides the system into four sub-systems: three access sub-systems (one for each CE level), and one NPUSCH transmission sub-system. As shown in Figure 2, the arrivals to the system are distributed among the three access sub-systems according to the RSRP thresholds. Each access sub-system consists of two tandem queues: the first one corresponds to NPRACH preamble detection, and the second one corresponds to *msg3* transmission. The NPRACH configuration $c^{(i)}$, together with δ, determine the service rate of sub-system i, which we refer to as CE i access rate, since it expresses the rate of UEs successfully accessing the system (*i.e.* establishing a connection). These three UE flows merge back into the NPUSCH transmition sub-system. The critical aspect is that all 4 subsystems share the same time-frequency resources in the uplink carrier.

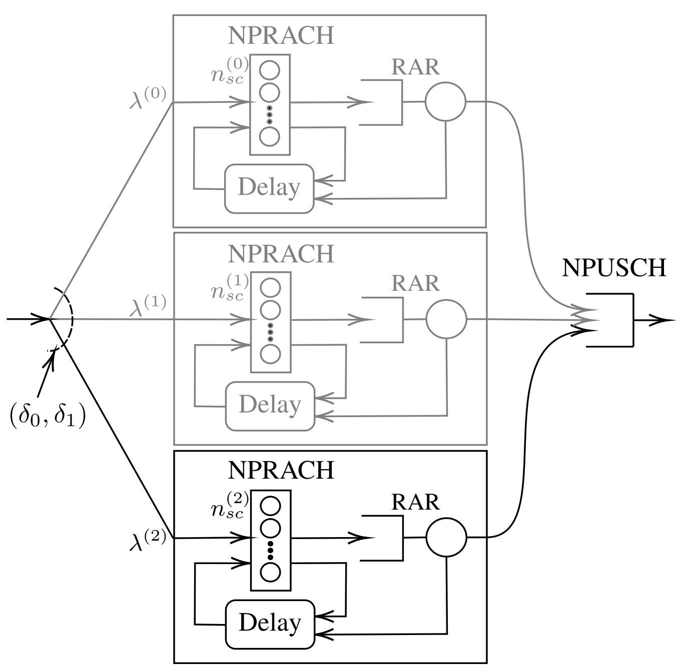


Fig. 2: Queueing models of the four sub-systems sharing the time-frequency resources of the uplink carrier. The RSRP thresholds (δ_0, δ_1) determine the arrival rate to each access sub-system. The UEs selecting a unique preamble in the NPRACH will continue to the RAR stage, where those whose *msg3* is correctly received become connected and proceed to the NPUSCH sub-system where they wait for an uplink grant in a common queue.

Considering the above model, our proposal estimates the UE arrival rate and the access rates of each CE level to configure each NPRACH as efficiently as possible. The higher the efficiency, the more radio resources are available for NPUSCH transmissions.

Let $A^{(i)}(t)$ be the number of CE level i UEs that have attempted to access the system up to instant t. We define the arrival rate at CE level i as

$$\lambda^{(i)} = \lim_{t\to\infty} \frac{A^{(i)}(t)}{t} \tag{3}$$

And let $D^{(i)}(t)$ be the number of UEs that have gained access to the system up to instant t. We define the access rate for CE level i as

$$\mu^{(i)} = \lim_{t\to\infty} \frac{D^{(i)}(t)}{t} \tag{4}$$

Assuming that the system has sufficient capacity, the configuration of the NPRACH resources must guarantee that all UEs trying to access the system succeed. In other words, in terms of queuing theory, we must guarantee that

$$\mu^{(i)} > \lambda^{(i)} \text{ for } i = 0, 1, 2 \tag{5}$$

These quantities are determined by the control variables δ and $c^{(i)}$ for $i = 0, 1, 2$. When the sub-system operates under condition (5) we say that it is stable. Ensuring stability involves several difficulties. For example, $\lambda^{(i)}$ must be estimated from incomplete observations (e.g. detections and collisions in each NPRACH) and during a limited observation time (the SIB2 period). Likewise, the estimation of $\mu^{(i)}$ requires simplifications that may entail additional inaccuracies.

Since δ distributes the arrivals among the sub-systems, we use $\widehat{\lambda}^{(i)}(\delta)$ to refer to the estimated $\lambda^{(i)}$ for a given δ. This estimate depends on the observation S_t, which we omit for the sake of clarity. To estimate $\mu^{(i)}$ we use a function $\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)$, detailed in the next subsection, that takes δ and the NPRACH configuration parameters ($n_{sc}^{(i)}$, $p^{(i)}$) as inputs. Considering the estimated variables, we replace the normalization condition (5) by the following one

$$\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) \geq \beta\widehat{\lambda}^{(i)}(\delta) \text{ for } i = 0, 1, 2 \tag{6}$$

where $\beta > 1$ is a configurable parameter with three interpretations: First, β acts as an additional slack factor for the access rate with respect to the estimated arrival rate, compensating for inaccuracies in the estimates that may result in under-provisioning of resources and thus instability. Second, $1/\beta$ acts as an upper bound on the load of each access sub-system, *i.e.*:

$$\frac{1}{\beta} \geq \frac{\widehat{\lambda}^{(i)}(\delta)}{\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)} \approx \rho^{(i)} \text{ for } i = 0, 1, 2 \tag{7}$$

Consequently, if each sub-system operates with $\rho^{(i)}$ close to this upper bound, a properly tuned β should attain the optimal balance between delay reduction (small $\rho^{(i)}$) and low resource occupation (large $\rho^{(i)}$). And third, a far-sighted agent (*e.g.* an RL agent), and thus capable of anticipating increments or decrements of the UE arrival intensity, can reflect this forecast in β allowing the system to adjust its access rate to the expected arrivals during the upcoming SIB2 period. Figure 3 summarizes the general structure of our model-based proposal.

The configuration problem consists of guaranteeing the access of as many incoming UEs as possible assigning the

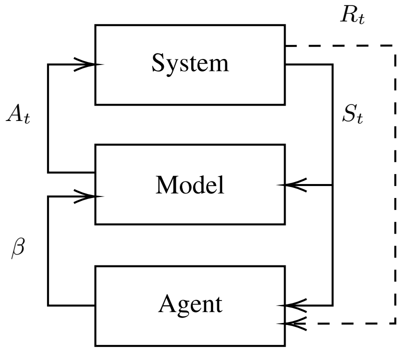


Fig. 3: Based on S_t, the agent determines the parameter β, which the model uses in combination with S_t to generate the control action A_t. The agent uses the reward R_t to update the control policy.

minimum amount of time-frequency resources from the carrier to NPRACHs. Defining the function $r^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)$ as the ratio of resources consumed by the access sub-system i, we can formulate the configuration problem as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=0}^{2} r^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) \\ \text{subject to} \quad & \mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) \geq \beta \widehat{\lambda}^{(i)}(\delta), \text{for } i = 0, 1, 2. \end{aligned} \tag{8}$$

where the optimization variables are δ, and ($n_{sc}^{(i)}$, $p^{(i)}$) for $i = 0, 1, 2$.

B. Elements of the Model

1) Estimation of the incoming traffic $\lambda^{(i)}(\delta)$: The estimation of the incoming traffic relies on a result from [25], which provides the following recursive expression for estimating the probability of d preambles being selected by exaclty one UE, and c preambles being selected by two or more UEs, when n UEs choose among r preambles:

$$\begin{aligned} P_n(d, c, r) = & \frac{r - (d - 1 + c)}{r} P_{n-1}(d - 1, c, r) \\ & + \frac{d + 1}{r} P_{n-1}(d + 1, c - 1, r) + \frac{c}{r} P_{n-1}(d, c, r), \end{aligned} \tag{9}$$

where, $P_0(0, 0, r) = 1$, and $P_n(d, c, r) = 0$ if $(d, c) \notin R_n$, defined as $R_n = \{(d, c) \in \mathbb{N}_0 \times \mathbb{N}_0 | d + c \leq r; d + \alpha c = n, \alpha \geq 2\}$. During the SIB2 update period, *i.e.* between two configuration stages, the controller observes the outcomes of $N^{(i)}$ NPRACH instances for CE level i, consisting on the number of detected ($d_\tau^{(i)}$) and collided ($c_\tau^{(i)}$) preambles for $\tau = 1, \ldots, N^{(i)}$. For each NPRACH instance we can use a maximum likelihood estimator (see [25]) of the arrivals:

$$\widehat{a}_\tau^{(i)} = \arg\max_n \; P_n(d_\tau^{(i)}, c_\tau^{(i)}, n_{sc}^{(i)}). \tag{10}$$

and obtain the estimation of the arrival rates (in arrivals per subframe) during the last observation period as

$$\widehat{\lambda}^{(i)} = \frac{1}{N^{(i)}} \sum_{\tau=1}^{N^{(i)}} \frac{\widehat{a}_\tau^{(i)}}{p^{(i)}}, \text{ for } i = 0, 1, 2 \tag{11}$$

The aggregate arrival rate estimation is then $\widehat{\lambda} = \sum_{i=0,1,2} \widehat{\lambda}^{(i)}$, which allows us to estimate the arrival rates at each CE for a given δ:

$$\widehat{\lambda}^{(i)}(\delta) = \widehat{\lambda} h^{(i)}(\delta), \text{ for } i = 0, 1, 2 \tag{12}$$

where $h^{(i)}(\delta)$ represents the fraction of users belonging to CE level i, and is obtained from the estimated RSRP distribution h as follows:

$$h^{(i)}(\delta) = \sum_{k \in \mathcal{L}_i(\delta)} h(l_k), \text{ for } i = 0, 1, 2 \tag{13}$$

where $\mathcal{L}_i(\delta)$ denotes the set of RSRP intervals in CE level i:

$$\mathcal{L}_0(\delta) = \{k : \delta_0 < l_k\} \tag{14}$$

$$\mathcal{L}_1(\delta) = \{k : \delta_1 < l_k \leq \delta_0\} \tag{15}$$

$$\mathcal{L}_2(\delta) = \{k : l_k \leq \delta_1\} \tag{16}$$

Efficient implementation: In terms of computational overhead, the critial part of the above procedure is the estimation of $\widehat{a}_\tau^{(i)}$ at each NPRACH period with (10), given the values of $d_\tau^{(i)}$, $c_\tau^{(i)}$, and $n_{sc}^{(i)}$. Note that $n_{sc}^{(i)}$ can only take values from a very small set (*e.g.* $n_{sc}^{(i)} \in \{12, 24, 36, 48\}$ for a single carrier system) and the pair ($d_\tau^{(i)}$, $c_\tau^{(i)}$) can only take values from the set $\{(d_\tau^{(i)}, c_\tau^{(i)}) \in \mathbb{N}_0 \times \mathbb{N}_0 : d_\tau^{(i)} + c_\tau^{(i)} \leq n_{sc}^{(i)}\}$ containing $(n_{sc}^{(i)} + 1)n_{sc}^{(i)}/2$ elements. Consequently, the number of possible combinations of $d_\tau^{(i)}$, $c_\tau^{(i)}$, and $n_{sc}^{(i)}$, in a single carrier system, is $13 \times 12/2 + 25 \times 24/2 + 37 \times 36/2 + 49 \times 48/2 = 2220$. Therefore, it is feasible to pre-compute the estimator (10) offline for all these combinations, and store them in a lookup table ($A^\star$), such that $\widehat{a}_\tau^{(i)} = A^\star(d_\tau^{(i)}, c_\tau^{(i)}, n_{sc}^{(i)})$, thus trading computational effort for memory storage. In online operation, the only required computations are $\widehat{\lambda}^{(i)}(\delta)$ (11), bounded by the maximum number of NPRACH periods within a SIB2 period, and $h^{(i)}(\delta)$ (13), bounded by the number of RSRP threshold values (12 in our example).

2) Estimation of the access rate μ: To estimate the maximum number of UE access detections per NPRACH we need to consider, for each CE, the maximum number of *msg3* messages, $m_{\text{max}}^{(i)}$, that can be allocated within the RAR window *before the start of the next NPRACH* resource of this CE. Thus we need to consider the length of the RAR window and the NPRACH period for each CE. The duration of the RAR window is determined by the *ResponseWindowSize* parameter (w_{RAR}), given in NPDCCH periods. Between two consecutive NPRACH resources, there are up to $p^{(i)} - n_{sf}^{(i)}(\delta) - 4$ subframes available for *msg3* transmissions, where $n_{sf}^{(i)}(\delta)$ denotes the number of subframes of the NPRACH from which we must subtract a guard time interval of 4 subframes. Therefore the available number of NPDCCH periods for RAR signaling in CE level i is:

$$w_{RAR}^{(i)} = \max\left(\left\lfloor \frac{p^{(i)} - n_{sf}^{(i)}(\delta) - 4}{p_{\text{DCCH}}} \right\rfloor, w_{RAR} \right) \tag{17}$$

where p_{DCCH} denotes the NPDCCH period length in subframes. We must determine the available capacity in DCIs for sending *msg2* messages, and the available capacity in subframes of the uplink carrier for sending *msg3* messages.

Denoting by n_{DCI} the number of DCIs that fit into each NPDCCH, there are $w_{\text{DCI}}^{(i)} = n_{\text{DCI}} w_{RAR}^{(i)}$ DCIs, and $w_{\text{RU}}^{(i)} = p_{\text{DCCH}} w_{RAR}^{(i)}$ subframes per NPRACH period. For CE level i, the number of DCI repetitions for *msg2* is denoted by $n_{\text{DCI}}^{(i)}$, and the number of RUs of a *msg3* is denoted by $n_{RU}^{(i)}$, thus the limit $m_{\text{max}}^{(i)}$ imposed by the RAR window capacity is given by

$$m_{\text{max}}^{(i)} = \min \left(\left\lfloor \frac{w_{\text{RU}}^{(i)}}{n_{RU}^{(i)}} \right\rfloor , \left\lfloor \frac{w_{\text{DCI}}^{(i)}}{n_{\text{DCI}}^{(i)}} \right\rfloor \right) \tag{18}$$

where the dependency on $p^{(i)}$ and δ is omitted for the sake of clarity.

The maximum expected number of UEs gaining access per NPRACH period in CE level i is given by:

$$d_i^\star(n_{sc}^{(i)}, p^{(i)}, \delta) = \max_n \sum_{d=1}^{n_{sc}^{(i)}} \min(d, m_{\text{max}}^{(i)}) P_n(d, n_{sc}^{(i)}) \tag{19}$$

where $P_n(d, n_{sc}^{(i)})$ denotes the probability of d preamble detections when n UEs choose among $n_{sc}^{(i)}$ preambles, given by the marginal distribution of (9) over collisions:

$$P_n(d, n_{sc}^{(i)}) = \sum_{c=0}^{n-d} P_n(d, c, n_{sc}^{(i)}) \tag{20}$$

The access rate function is defined as follows

$$\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) = \frac{d_i^\star(n_{sc}^{(i)}, p^{(i)}, \delta)}{p^{(i)}} e^{(i)}(\delta) \tag{21}$$

where $e^{(i)}(\delta)$ denotes the *msg3* detection efficiency. To estimate $e^{(i)}(\delta)$, let us define $e_k^{(i)}(\delta)$ as the pre-estimated detection probability of a *msg3* from a device in the k-th RSRP interval, using the *msg3* configuration determined by δ for CE level i. Then, the expected detection rate for users in CE level i can be estimated with h:

$$e^{(i)}(\delta) = \frac{\sum_{k \in \mathcal{L}_i(\delta)} e_k^{(i)}(\delta) h(l_k)}{h^{(i)}(\delta)}, \text{ for } i = 0, 1, 2 \tag{22}$$

with $\mathcal{L}_i(\delta)$ given by (14, 15, 16).

Efficient implementation: Obtaining $d_i^\star(n_{sc}^{(i)}, p^{(i)}, \delta)$ (19) is the critical step in terms in computational complexity. Similarly to previous subsection, we define a lookup table $D^\star$ storing the pre-computed values of $d_i^\star$ for all the possible combinations of i, $n_{sc}^{(i)}$, $p^{(i)}$, and δ. Consider a system with 12 RSRP threshold values, resulting in $13 \times 12/2 = 78$ values of δ, 12 of which correspond to 2 CE levels, and 66 to 3 CE levels. Considering 4 possible values for $n_{sc}^{(i)}$ and 6 values for $p^{(i)}$, the storage requirement of the lookup table for $D^\star$ is $3 \times 66 \times 24 + 2 \times 12 \times 24 = 5328$ entries. During online operation, the only required computation is $e^{(i)}(\delta)$ (22), bounded by the number of RSRP threshold values (12 in our example).

3) Estimation of the resource occupation $r^{(i)}$: The resource occupation function $r^{(i)}$ provides the ratio between resources devoted to RA (NPRACH and *msg3* transmissions) for CE i, and the total uplink carrier resources. Given δ, $n_{sc}^{(i)}$ and $p^{(i)}$, the NPRACH occupies $n_{sf}^{(i)}(\delta) n_{sc}^{(i)}$ time-frequency resources out of the $48p^{(i)}$ resources available during a NPRACH period lasting $p^{(i)}$ subframes. During that period, *msg3* messages are transmitted at the access rate $\mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta)$, in messages per subframe, consuming $n_{RU}^{(i)}(\delta)$ resources, which is equivalent to one subframe in terms of time-frequency resources. Therefore we have:

$$r^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) = \frac{n_{sf}^{(i)}(\delta) n_{sc}^{(i)}}{48p^{(i)}} + \mu^{(i)}(n_{sc}^{(i)}, p^{(i)}, \delta) n_{RU}^{(i)}(\delta) \tag{23}$$

C. Efficient Solution Strategy

For a given δ, the configuration problem (8) is separable into 3 sub-problems defined as:

$$\begin{aligned} \underset{c^{(i)}}{\text{minimize}} \quad & r^{(i)}(c^{(i)}, \delta) \\ \text{subject to} \quad & \mu^{(i)}(c^{(i)}, \delta) \geq \beta \widehat{\lambda} h^{(i)}(\delta) \end{aligned} \tag{24}$$

for $i = 0, 1, 2$, where $c^{(i)} = (n_{sc}^{(i)}, p^{(i)})$ represents the NPRACH configuration for CE i. There exists a solution to (24) provided the following condition holds:

$$\beta \widehat{\lambda} h^{(i)}(\delta) \leq \mu_{max}^{(i)}(\delta) \tag{25}$$

where $\mu_{max}^{(i)}(\delta)$ denotes the maximum achievable access rate for CE i given δ. Note that $h^{(i)}(\delta)$ represents the fraction of incoming traffic going to sub-system i given the RSRP thresholds defined by δ. Therefore, (25) implies that there exists a stable configuration for sub-system i. Let $\Delta_f(\beta\widehat{\lambda})$ denote the subset of δ values in Δ fulfilling (25) for $i = 0, 1, 2$

When the feasibility condition (25) holds, we define $c_i^\star(\delta, \beta\widehat{\lambda})$ as the solution of (24) for a given δ and $\beta\widehat{\lambda}$. Otherwise, $c_i^\star(\delta, \beta\widehat{\lambda})$ equals the rate maximizing configuration (*i.e.* $\mu^{(i)}\big(c_i^\star(\delta, \beta\widehat{\lambda}), \delta\big) = \mu_{max}^{(i)}(\delta)$).

The optimal δ is given by

$$\underset{\delta \in \Delta_f(\beta\widehat{\lambda})}{\text{minimize}} \left[\sum_{i=0}^{2} r^{(i)}\big(c_i^\star(\delta, \beta\widehat{\lambda}), \delta\big) \right] \tag{26}$$

If one or more sub-systems do not fulfill the feasibility condition (25), *i.e.* if $\Delta_f(\beta\widehat{\lambda}) = \varnothing$, the system is considered to be under heavy incoming traffic. Then, the objective becomes to bring the access and the arrival rates as close as possible:

$$\underset{\delta}{\text{minimize}} \left[\sum_{i=0}^{2} \Big(\mu^{(i)}\big(c_i^\star(\delta, \beta\widehat{\lambda}), \delta\big) - \beta\widehat{\lambda} h^{(i)}(\delta) \Big)^2 \right] \tag{27}$$

Because of the low mobility of NB-IoT devices, we can expect negligible variations on the RSRP distribution estimation h after a sufficient amount of samples. To allow for a more efficient implementation of the configuration problem, we leverage the fact that, once h is stabilized, the only variable factor is the product $\beta\widehat{\lambda}$. Thus, instead of solving (26, 27) at each decision step, it is much more computationally efficient to define a lookup table $C^\star$ mapping a set of pre-defined values of $\beta\widehat{\lambda}$, denoted by $(\beta\widehat{\lambda})_0, (\beta\widehat{\lambda})_1, \ldots$, to optimal configurations. These $(\beta\widehat{\lambda})_j$ values are evenly spaced, ranging from $\beta_{min}\widehat{\lambda}_{min}$ to $\beta_{max}\widehat{\lambda}_{max}$, where $(\beta_{min}, \beta_{max})$ and $(\widehat{\lambda}_{min}, \widehat{\lambda}_{max})$ denote the extreme ranges for β and $\widehat{\lambda}$ respectively. At each decision stage, $\beta\widehat{\lambda}$ is approximated by its nearest value $(\beta\widehat{\lambda})_j$, and is used to retrieve the optimal configuration if it was already computed from a previous stage. If not, it is computed using the previously described procedure and stored for later reuse. The memory requirement for $C^\star$ is limited to the number of $\beta\widehat{\lambda}$ values defined.

D. The Model-Based Configuration

The model-based selection of configuration operates in two phases. The first one, summarized in Algorithm 1, corresponds to the initialization of the estimated distribution h. During this phase, the model operates with a predefined δ configuration, and obtains the NPRACH parameters by solving (26) for the given δ. The initialization phase ends when h has been updated with at least n_{init} RSRP samples. The variable β_t refers to the value of the controlled parameter β generated by the control agent at time t. After the initialization phase, the model operates according to Algorithm 2.

Algorithm 1 **Model initialization**

Inputs: δ, n_{max}, lookup tables $A^\star$, $D^\star$
$n = 0$ ▷ arrivals counter
while $n < n_{\text{init}}$ **do**
  for $t = 1, 2, \ldots$ **do**
    Receive observation S_t and control variable β_t.
    Update n and h
    Estimate $\widehat{\lambda}$ with $A^\star$ and (11)
    Obtain ($c_0^\star$, $c_1^\star$, $c_2^\star$) by solving (24) for $\beta_t\widehat{\lambda}$
    return (δ, $c_0^\star$, $c_1^\star$, $c_2^\star$)
  end for
end while

Algorithm 2 **Model-based configuration selection**

Inputs: h, lookup tables $A^\star$, $D^\star$, and $C^\star$ (empty)
for $t = 1, 2, \ldots$ **do**
  Receive observation S_t and control variable β_t.
  Estimate $\widehat{\lambda}$ with $A^\star$ and (11)
  Find $(\beta\widehat{\lambda})_j$ closest to $\beta_t\widehat{\lambda}$
  if $(\beta\widehat{\lambda})_j$ in $C^\star$ **then**
    ($\delta^\star$, $c_0^\star$, $c_1^\star$, $c_2^\star$) $\leftarrow C^\star[(\beta\widehat{\lambda})_j]$ ▷ retrieve
  else
    if $\Delta_f(\beta\widehat{\lambda}) \neq \varnothing$ **then**
      Obtain ($\delta^\star$, $c_0^\star$, $c_1^\star$, $c_2^\star$) by solving (24, 26)
    else
      Obtain ($\delta^\star$, $c_0^\star$, $c_1^\star$, $c_2^\star$) by solving (24, 27)
    end if
    $C^\star[(\beta\widehat{\lambda})_j] \leftarrow$ ($\delta^\star$, $c_0^\star$, $c_1^\star$, $c_2^\star$) ▷ store
  end if
  return ($\delta^\star$, $c_0^\star$, $c_1^\star$, $c_2^\star$)
end for

Solving (26) requires solving, for each $\delta \in \Delta$, the 3 subproblems (24). In the worst case, solving each subproblem implies evaluating all configurations $c^{(i)} = (n_{sc}^{(i)}, p^{(i)})$ for the given δ. Considering the previous example, finding the optimal configuration for a given $(\beta\widehat{\lambda})_j$ value requires, at most, 5328 queries to $D^\star$, the first time this value is encountered, and just 1 query to $C^\star$ afterwards.

IV. Evaluation

A. Methodology

1) Simulation Environment: The proposal was evaluated using an NB-IoT simulation environment, created using Python[1], The main elements of the simulator are: 1) a population of devices randomly distributed in an hexagonal cell, attempting to access the system for transmissting their data packets, 2) a base station, located in one edge of the hexagonal cell, that manages the access procedure and arranges transmission opportunities for the devices, 3) one or more carriers that allocate resources for NPRACH and NPUSCH, and 4) the channel models for NPRACH and NPUSCH transmissions.

Devices are idle until they become active according to a probabilistic traffic model. In particular, we use the two mMTC traffic models defined in 3GPP TR 37.868 [40]. Traffic model 1 considers time periods of $T = 60$ seconds over which each device attempts to access the network with a uniform probability. Traffic model 2 defines shorter periods of time $T = 10$ seconds, such that the number of users becoming active during a time interval defined by $t_1 < t_2 \leq T$ is given by

$$\text{arrivals} = N \int_{t_1}^{t_2} p(t)dt \tag{28}$$

where N is the number of (model 2) devices in the cell, and $p(t)$ is the Beta distribution given by:

$$p(t) = \frac{t^{a-1}(T-t)^{b-1}}{T^{a+b-1}B(a,b)} \tag{29}$$

where $a > 0$, $b > 0$ and $B(a,b)$ is the Beta function.

We define three **scenarios** according to the traffic:

1) *uniform*, in which all users follow traffic model 1.
2) *mixed*, where UE traffic is split between model 1 and model 2 with equal probability
3) *random*. In this scenario, at the start of each period of duration T, a random percentage of users, up to a maximum of 80%, will follow traffic model 2 during this period, while the rest will follow model 1.

The environment employs the propagation conditions and antenna patterns outlined in sections 4.2 and 4.5 of 3GPP TR 36.942 [41], respectively. It incorporates a block fading model in which a channel realization remains constant for each (NPRACH or NPUSCH) transmission and varies independently from one transmission to the next, following a lognormal shadow fading model with a standard deviation of $\sigma = 8$ dB. The identification of a preamble sequence relies on the probability model provided by [42], and the detection of NPUSCH transmissions employs the block error rate tables provided in [43] for each link-adaptation parameter setting.

The control agent selects the RSRP thresholds (δ_0 δ_1), and the NPRACH parameters ($p^{(i)}$ and $n_{sc}^{(i)}$ for $i = 0, 1, 2$). The values of these controlled parameters are shown in Table II.

[1]The simulator's code, the proposed algorithm, and the scripts needed for experiment replication can be found at https://github.com/jjalcaraz-upct/nb-iot-environment

Parameter (Unit)	Values
δ_i (dBm)	-116, -115, -114, -113, -112, -110, -108, -106, -104, -102, -100, -98
$p^{(i)}$ (ms)	80, 160, 240, 320, 640, 1280
$n_{sc}^{(i)}$ (preambles)	12, 24, 36, 48

TABLE II: Possible values of the controlled parameters.

Note that the selected (δ_0, δ_1) pair must fulfill $\delta_1 \leq \delta_0$, and when $\delta_1 = \delta_0$ only two CE levels are defined in the cell (CE2 and CE1). The total number of possible configurations are $66 \times 6^3 \times 4^3 + 12 \times 6^2 \times 4^2 = 919296$, which is a nontrivial amount for an RL algorithm.

The number of repetitions, for CE level 2, is set to $n_{rep}^{(2)} = 8$, and for CE levels $i = 0, 1$ $n_{rep}^{(i)}$ is determined by δ_i as shown in Table III.

Range	$\mathbf{n_{rep}^{(i)}}$
$\delta_i \leq -115$	8
$-115 < \delta_i \leq -112$	4
$-112 < \delta_i \leq -108$	2
$-108 < \delta_i$	1

TABLE III: Preamble repetitions for CE level i given δ_i.

These values provided a detection probability higher than 0.98 in our environment. Similarly, the link-adaptation parameters for *msg3* tansmissions in CE level i are predetermined according to the δ_i configuration. In particular, each *msg3* transmission uses one resource unit and QPSK modulation in all CE levels, and differs in the number of repetitions. For CE level 2, this number is set to 16, and for CE levels 1 and 0 the *msg3* repetitions are given by Table IV.

Range	msg3 repetitions
$\delta_i \leq -113$	16
$-113 < \delta_i \leq -108$	8
$-108 < \delta_i \leq -104$	4
$-104 < \delta_i \leq -101$	2
$-101 < \delta_i$	1

TABLE IV: *msg3* repetitions for CE level i given δ_i.

The backoff time is set to 1024 ms or to 4096 ms, depending on the NPRACH periodicity. The remaining parameters are configured according to Table V.

The NPUSCH scheduling and transmission parameters are determined by a simple decision rule: The UE with the longest connection time is selected first, and HARQ retransmissions are prioritized. The link-adaptation parameters are automatically selected based on the reported pathloss value, the transport block size required to transmit its data buffer, and a target error rate of 10%. Other NPUSCH control approaches can be used, even policies learned by RL agents as in [35], but the interaction between RL agents learning different network functionalities concurrently is a challenge in itself that falls outside the scope of this paper.

2) RL Agents: The MBRL proposal combines the model described in section III with an RL agent that, at each time step t, receives the observation S_t from the system and generates a value for the β parameter, β_t, that is used by the model to generate action A_t. For the RL agent, we consider the following state-of-the-art deep RL algorithms.

Parameter	Value
Device transmission power	23 dBm
Base station antenna gain	15 dBi
Receiver noise figure	3 dB
Receiver effective noise power	-121 dBm
Reference signal receive power	35 dBm
Number of devices in the cell	1000
Hexagonal cell radius	5 Km
NPDCCH periodicity	30 ms
NPDCCH consecutive subframes	8
SIB2-NB update period	3 s
RAR window	8 NPDCCH periods
Contention timer	3 NPDCCH periods
Maximum access attempts	5
Length of a preamble sequence	5.6 ms (format 0)
Packet size	Between 100 and 600 bits
MBRL range of values for β	$[0.3, 2.0]$
MBRL initialization samples, n_{init}	1000

TABLE V: Parameter setting of the simulation environment.

- **Proximal policy optimization (PPO)** [44] is a model-free deep policy gradient algorithm that updates policies preventing the new policy to diverge too much with respect to the previous one, in order to avoid unstable behavior during the learning process.
- **Synchronous advantage actor critic (A2C)** [45] is an on-policy deep actor-critic algorithm that uses the advantage function to evaluate actions. The term *synchronous* indicates that it can execute multiple instances of the algorithm in parallel, but this feature is not applicable in online learning, where only one instance of the environment is available.
- **Soft Actor Critic (SAC)** [46] is an off-policy deep actor-critic algorithm that optimizes a stochastic policy in an entropy-augmented reward framework, leading to more exploratory and robust policy learning. It is designed for continuous action spaces and combines the benefits of actor-critic methods with those of maximum entropy reinforcement learning.

As **baselines**, we use two of the above algorithms, PPO and A2C, operating without the model, *i.e.* as model-free RL (MFRL) agents. Model-free SAC is not included as a baseline since it is not suitable for discrete action spaces. We use the RL implementations provided by *Stable Baselines 3* [47], which is an improved version of the OpenAI Baselines [48]. Other RL algorithms were evaluated: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), for MBRL, and Deep Q-Networks (DQN) for MFRL, but their results are not included because of their poor performance compared to the previous ones.

3) Evaluation Experiments: Each (MBRL or MFRL) agent has been evaluated in 20 independent simulation runs on each scenario. Each simulation run consists of a 100000-step online learning episode in which the agent starts with no prior knowledge and learns over time. We consider three metrics:

1) the *departures* defined as the number of UEs that have successfully completed a NPUSCH transmission between consecutive decision steps,
2) the *service time*, defined as the time elapsed from the moment a UE initiates the access process until it

completes its NPUSCH transmission, and
3) the *NPRACH resources*, defined as the ratio of carrier resources devoted to NPRACH.

We estimate the average of each metric over 20 runs, and its confidence interval with a confidence level of 90%.

B. Numerical Results

1) Uniform Scenario: Figure 4 shows how the average number of *departures* per step evolves over the 100000 decision steps in the *uniform* scenario for the evaluated algorithms. The MBRL agents are identified by "MB" followed by the name of the RL algorithm. We found that all agents converge to a similar value, with MFRL agents converging at a slower rate, especially PPO. To illustrate this effect more clearly, the figure includes two additional plots, one zooming in on earlier stages (the first 500 steps), and the other on later stages (from step 99000 to step 99500). We see that the MB algorithms outperform MF ones from the beginning, and keep improving during the first 20000 steps.

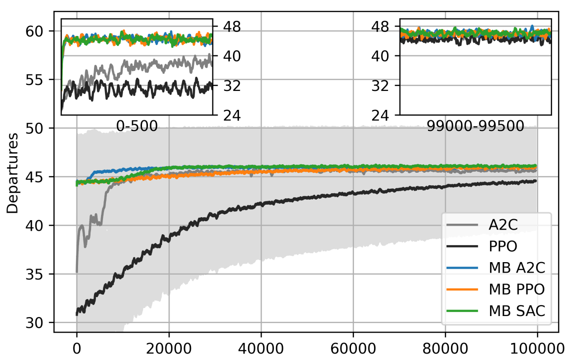


Fig. 4: Average *departures* per time step during the learning episodes of each algorithm in the *uniform* traffic scenario.

The difference in performance is more pronounced in terms of *service time*, as shown by Figure 5, where we observe that the mean service times of MB A2C and MB SAC, go from around 8 seconds to close to 5 seconds during the first 20000 steps. The MF algorithms converge to a similar delay, but their mean service times are above 15 seconds (A2C) and 30 seconds (PPO), during the first 10000 steps. It is important to take into account the time scale of the episodes. Since each stage corresponds to a SIB2-NB update occurring every 3 seconds, the duration of each training episode is 300000 seconds, which corresponds to 3 days, 11 hours and 20 minutes. The zoomed in period of 500 steps corresponds to 25 minutes. From this perspective, the benefit of using MBRL for online learning is evident.

To perform a more accurate comparison of the *service time* in different stages of the learning episodes, Figure 6 shows the estimated distributions of the *service times* attained by each algorithm at the beginning, the middle, and at the end of the episodes. The initial phase comprises the first 5000 steps, (*i.e.* it lasts 4 hours and 10 minutes). The intermediate phase ranges from 45000 to 50000 steps (*i.e.* it starts after 1 day, 13 hours and 30 minutes), and the final phase from 95000 to 100000 steps (*i.e.* it starts after 3 days, 7 hours, and 10 minutes). We see that the differences in *service time* between MFRL and MBRL agents are very pronounced in the initial stage, and become smaller over time. It is worth noting that to approach the performance of the MB algorithms, the MF algorithms need more than a day and a half without changes in the environment, which is not always feasible. In the event of a significant change in the environment, the MB algorithms could simply restart the learning process and attain efficient performance in minutes.

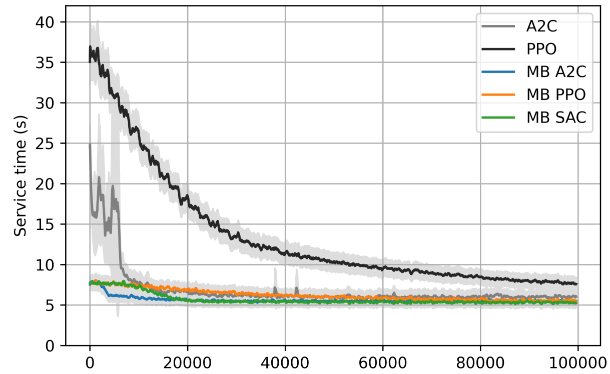


Fig. 5: Average *service time* measured per time step during the learning episodes of each algorithm in the *uniform* traffic scenario.

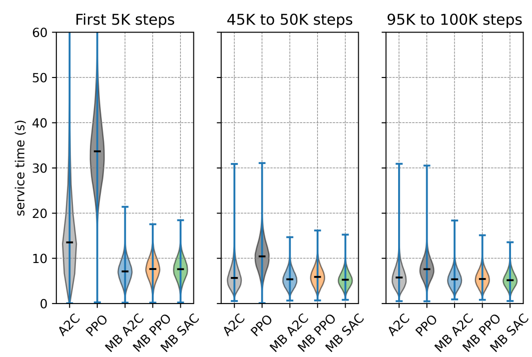


Fig. 6: Estimated distributions of the *service times* measured during different periods within the learning episodes of each algorithm in the *uniform* traffic scenario.

Figure 7 shows the evolution of the *NPRACH resources* ratio. Interestingly, MFRL agents converge to MBRL agents in this metric, which indicates that their learne policies are similar to those of MBRL agents in terms of resource allocation, and validates to some extent the assumptions made to develop the model. Finally, Figure 8 shows the evolution of the control parameter β selected by the MBRL agents over time. We see that in all three algorithms, the average β value lies between 0.6 and 0.8 in the long term, which suggests that the model either slightly overestimates the incoming traffic, or underestimates the access rate, or both.

2) Bursty Traffic Scenarios: Figure 9 shows the outputs for the *random* scenario, with two zoom windows on the initial and final stages. The zoom allows us to observe the

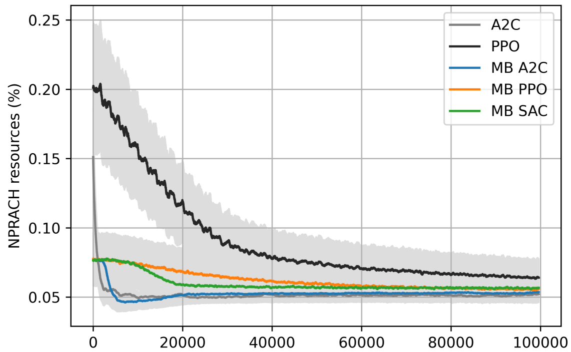


Fig. 7: Ratio of *NPRACH resources* over carrier resources during the learning episodes of each algorithm in the *uniform* traffic scenario.

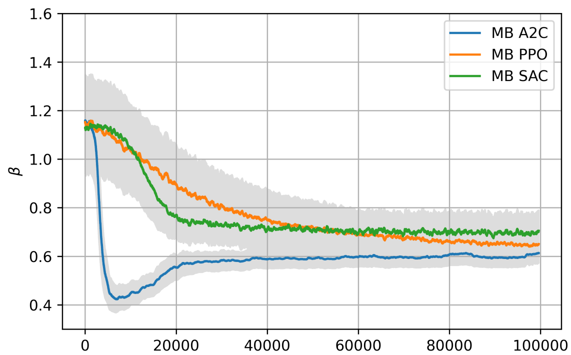


Fig. 8: Evolution of the control parameter β selected by the MBRL agents during the learning episodes in the *uniform* traffic scenario.

bursty nature of the departures, replicating the arrival pattern, which cannot be perceived in the main figure due to the sliding window averaging. In the final stages, all the algorithms bahave similarly in terms of departures per step. The results for the *mixed* scenario are very similar, and are omitted to avoid redundancy.

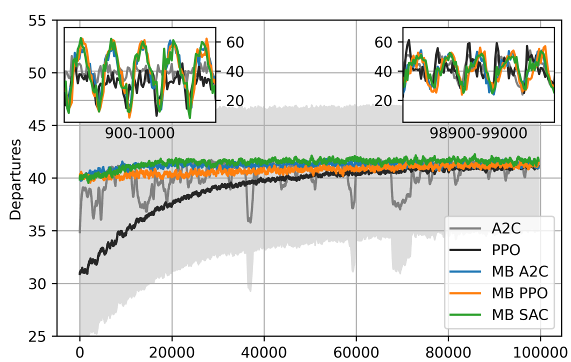


Fig. 9: Average *departures* per time step during the learning episodes of each algorithm in the *random* traffic scenario.

Although A2C approaches MB agents in terms of departures per time step, its frequent drops in performance raise concerns about its stability. In Figure 10, we can assess that these oscillations correspond to spikes in *service time*. The zoom of this figure shows that this metric also reflects the periodic nature of the traffic. To complete this evaluation, Figure 11 depicts the distributions of the *service time* in the initial, intermediate and final stages of the episodes. Again, we observe that the MB algorithms perform better than their MF counterparts at the beginning of the episode, and slightly better at the end.

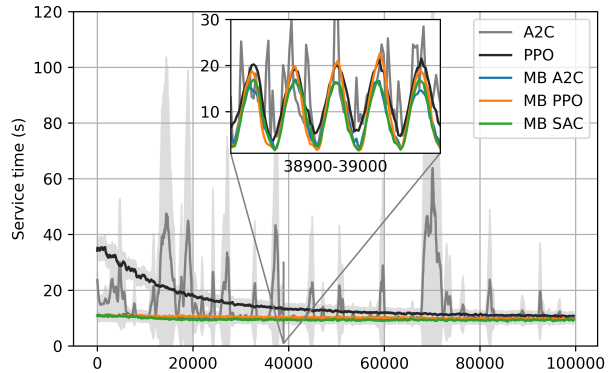


Fig. 10: Average *service time* measured per time step during the learning episodes of each algorithm in the *random* traffic scenario.

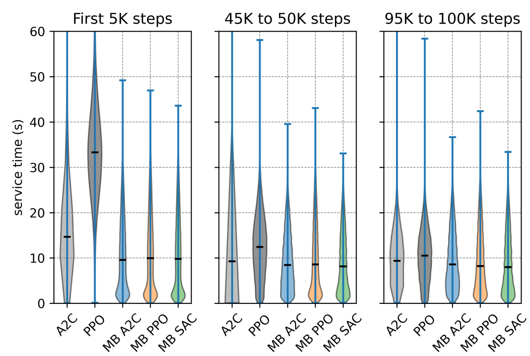


Fig. 11: Estimated distributions of the *service times* measured during different periods within the learning episodes of each algorithm in the *uniform* traffic scenario.

Figure 12 shows the evolution of the *NPRACH resources* ratio during learning, where we see how the algorithms converge to policies that assign approximately 5% of the carrier resources (7% in the case of PPO) to NPRACH. Finally, Figure 13 shows the average value of the control action β selected by the agents as the learning episode progresses. Zooming in on the earlier steps, we see how β still lacks structure, whereas in the later steps β follows the oscillatory pattern of the arrivals, which suggests that the RL agent learns a *predictive* policy, such that the resulting traffic estimation $(\widehat{\beta\lambda})$ anticipates the incoming traffic in the upcoming step.

V. Conclusion

This paper presents a novel approach for the automatic configuration of NPRACH parameters and the RSRP thresholds that define the coverage area for each CE level in

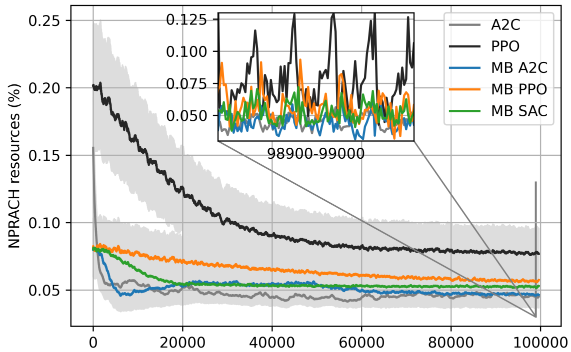


Fig. 12: Ratio of *NPRACH resources* over carrier resources during the learning episodes of each algorithm in the *random* traffic scenario.

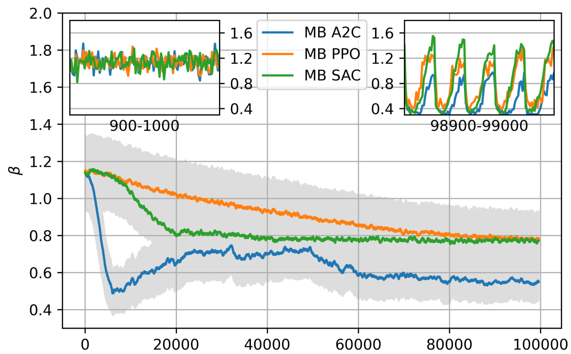


Fig. 13: Evolution of the control parameter β selected by the MBRL agents during the learning episodes in the *random* traffic scenario.

NB-IoT networks. Our proposal consists of a new MBRL algorithm that integrates classic network modeling techniques, such as queueing theory and combinatorial analysis into a parameterized model. In this architecture, the model and the agent operate together: both receive the observation of the environment, the agent adjusts the model parameter, and the model then generates the control action. The main advantage of using an MBRL approach is the significant increase in sample efficiency compared to the MFRL counterpart, enabling the agent to opeate online, making efficient decisions even during the early stages of learning. For instance, the average service time using the state-of-the-art PPO algorithm exceeds 30 seconds in the initial hours of operation, but with our MBRL scheme, it drops to less than 10 seconds. This improvement stems from embedding environmental dynamics into the model, offering a specialized solution for NB-IoT while also indicating that RL algorithms for network control can leverage decades of network modeling knowledge. A very promising future research line is integrating MBRL with automatic generation of models of the controlled system using generative artificial intelligence techniques.

References

[1] E. M. Migabo, K. D. Djouani, and A. M. Kurien, "The narrowband Internet of Things (NB-IoT) resources management performance state of art, challenges, and opportunities," *IEEE Access*, vol. 8, pp. 97 658–97 675, 2020.

[2] Y.-P. E. Wang, X. Lin *et al.*, "A primer on 3GPP narrowband Internet of Things," *IEEE communications magazine*, vol. 55, no. 3, pp. 117–123, 2017.

[3] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Physical Channels and Modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.211, 2022, version 17.2.0.

[4] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.212, 2022, version 17.1.0.

[5] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Physical Layer Procedures," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.213, 2022, version 17.4.0.

[6] M. Kanj, V. Savaux, and M. Le Guen, "A tutorial on NB-IoT physical layer design," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[7] L. Feltrin, G. Tsoukaneri *et al.*, "Narrowband IoT: A survey on downlink and uplink perspectives," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 78–86, 2019.

[8] R. Barbau, V. Deslandes *et al.*, "An Analytical Model for Assessing the Performance of NB-IoT," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[9] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[10] B.-Z. Hsieh, Y.-H. Chao, R.-G. Cheng, and N. Nikaein, "Design of a UE-specific uplink scheduler for narrowband Internet-of-Things (NB-IoT) systems," in *2018 3rd International Conference on Intelligent Green Building and Smart Grid (IGBSG)*. IEEE, 2018, pp. 1–5.

[11] O. Elgarhy, L. Reggiani *et al.*, "Rate-latency optimization for NB-IoT with adaptive resource unit configuration in uplink transmission," *IEEE Systems Journal*, vol. 15, no. 1, pp. 265–276, 2020.

[12] Y.-J. Yu and J.-K. Wang, "NPRACH-Aware Link Adaptation and Uplink Resource Allocation in NB-IoT Cellular Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4894–4906, 2021.

[13] Y.-J. Yu and L.-X. Li, "Offset-Aware Resource Allocation in NB-IoT Networks," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 967–23 980, 2022.

[14] Y.-J. Yu, "NPDCCH period adaptation and downlink scheduling for NB-IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 962–975, 2020.

[15] X. Chen, Z. Li, Y. Chen, and X. Wang, "Performance analysis and uplink scheduling for QoS-aware NB-IoT networks in mobile computing," *IEEE Access*, vol. 7, pp. 44 404–44 415, 2019.

[16] C. Yu, L. Yu *et al.*, "Uplink scheduling and link adaptation for narrowband Internet of Things systems," *IEEE Access*, vol. 5, pp. 1724–1734, 2017.

[17] R. Karmakar, G. Kaddoum, and S. Chattopadhyay, "SmartCon: Deep probabilistic learning-based intelligent link-configuration in narrowband-IoT toward 5G and B5G," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1147–1158, 2021.

[18] J. J. Alcaraz, F. Losilla, and F.-J. Gonzalez-Castaño, "Transmission Control in NB-IoT with Model-Based Reinforcement Learning," *IEEE Access*, 2023.

[19] C.-W. Huang, S.-C. Tseng, P. Lin, and Y. Kawamoto, "Radio resource scheduling for narrowband internet of things systems: A performance study," *IEEE Network*, vol. 33, no. 3, pp. 108–115, 2019.

[20] Y.-J. Yu, Y.-C. Wang, and C.-H. Fan, "Control Period Adaptation and Resource Allocation for Joint Uplink and Downlink in NB-IoT Networks," *IEEE Internet of Things Journal*, 2024.

[21] N. Jiang, Y. Deng, X. Kang, and A. Nallanathan, "Random access analysis for massive iot networks under a new spatio-temporal model: A stochastic geometry approach," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5788–5803, 2018.

[22] N. Jiang, Y. Deng *et al.*, "Analyzing random access collisions in massive IoT networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6853–6870, 2018.

[23] Y. Liu, Y. Deng, M. Elkashlan, and A. Nallanathan, "Random access performance for three coverage enhancement groups in NB-IoT networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[24] Y. Liu, Y. Deng *et al.*, "Analysis of random access in NB-IoT networks with three coverage enhancement groups: A stochastic geometry approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 549–564, 2020.

[25] L. Tello-Oquendo, V. Pla *et al.*, "Efficient random access channel evaluation and load estimation in LTE-A with massive MTC," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1998–2002, 2018.

[26] N. Jiang, Y. Deng, O. Simeone, and A. Nallanathan, "Online supervised learning for traffic load prediction in framed-ALOHA networks," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1778–1782, 2019.

[27] R. Harwahyu, R.-G. Cheng *et al.*, "Repetitions versus retransmissions: Tradeoff in configuring NB-IoT random access channels," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3796–3805, 2019.

[28] R. Harwahyu, R.-G. Cheng, D.-H. Liu, and R. F. Sari, "Fair configuration scheme for random access in NB-IoT with multiple coverage enhancement levels," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1408–1419, 2019.

[29] D. Pacheco-Paramo, L. Tello-Oquendo, V. Pla, and J. Martinez-Bauset, "Deep reinforcement learning mechanism for dynamic access control in wireless networks handling mMTC," *Ad Hoc Networks*, vol. 94, p. 101939, 2019.

[30] N. Jiang, Y. Deng, A. Nallanathan, and J. Yuan, "A decoupled learning strategy for massive access optimization in cellular IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 668–685, 2020.

[31] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[32] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[33] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NeurIPS*, Dec. 2018, pp. 4759–4770.

[34] L. Kaiser, M. Babaeizadeh *et al.*, "Model-based reinforcement learning for Atari," *arXiv preprint arXiv:1903.00374*, Feb. 2020.

[35] J. J. Alcaraz, F. Losilla, A. Zanella, and M. Zorzi, "Model-Based Reinforcement Learning With Kernels for Resource Allocation in RAN Slices," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 486–501, 2022.

[36] J. A. Ayala-Romero, J. J. Alcaraz, J. Vales-Alonso, and E. Egea-Lopez, "Online optimization of interference coordination parameters in small cell networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6635–6647, Oct. 2017.

[37] J. J. Alcaraz, J. A. Ayala-Romero, J. Vales-Alonso, and F. Losilla-López, "Online reinforcement learning for adaptive interference coordination," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 10, pp. 1–24, Aug. 2020.

[38] J. A. Ayala-Romero, J. J. Alcaraz, A. Zanella, and M. Zorzi, "Online Learning for Energy Saving and Interference Coordination in HetNets," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1374–1388, Jun. 2019.

[39] S. Riolo, D. Panno, and L. Miuccio, "Modeling and analysis of tagged preamble transmissions in random access procedure for mmtc scenarios," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4296–4312, 2021.

[40] 3GPP, "RAN Improvements for Machine-type Communications," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 37.868, 2011, version 11.0.0.

[41] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.942, 2018, version 15.0.0.

[42] J.-K. Hwang, C.-F. Li, and C. Ma, "Efficient detection and synchronization of superimposed NB-IoT NPRACH preambles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1173–1182, 2018.

[43] J. A. Z. Emmanuel Lujan, "Lite NBIoT NPUSCH Simulator," https://github.com/CSC-CONICET/Lite-NBIoT-NPUSCH-Simulator, 2019.

[44] J. Schulman, F. Wolski *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, Aug. 2017.

[45] V. Mnih, A. P. Badia *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, Jun. 2016, pp. 1928–1937.

[46] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[47] A. Raffin, A. Hill *et al.*, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[48] P. Dhariwal, C. Hesse *et al.*, "OpenAI Baselines," https://github.com/openai/baselines, 2017.

Juan J. Alcaraz obtained his MS degree in telecommunications engineering from the Polytechnic University of Valencia, Spain, in 1999, and his PhD in telecommunications engineering from the Technical University of Cartagena (UPCT), Spain, in 2007. He joined UPCT in 2004, where he is currently a Professor with the Department of Information and Communication Technologies. He was a Fulbright Visiting Scholar with the Electrical Engineering Department of the University of California, Los Angeles (UCLA), in 2013, and a visiting researcher with the Department of Information Engineering, University of Padova, in 2017 and in 2019. His current research focuses on learning algorithms for wireless networks.

Juan-Carlos Sanchez-Aarnoutse received the M.S. degree in Automation and Electronics Engineering (2002) and the Ph.D. degree in Telecommunications Engineering (2006) from the Technical University of Cartagena (UPCT), Spain. Since 2002, he is Assistant Professor in the Department of Information Technologies and Communications at the Universidad PolitÃ©cnica de Cartagena (Spain). He has authored several papers, book chapters, and conference proceedings in the areas of multicast protocols and P2P traffic measurement, as well as smart grid and power line communications and network capacity estimation. In the last years, he has been working in the field of IoT, UAVs and their applications.

Alejandro S. Martinez-Sala received the M.S. degree in Automation and Electronics Engineering in 2000 and the Ph.D. in Telecommunications from the Technical University of Cartagena (UPCT), Spain, in 2006. In 2001 he joined the UPCT where he is assistant professor at the Communications and Information Technologies Department. His research interests include IoT technology, location systems and services, and innovation management and technology transfer to industry.

Francisco-Javier Gonzalez-Castaño is a Full Professor with the Telematics Engineering Department, University of Vigo, Spain, where he leads the Information Technology Group. He has authored more than 100 papers in international journals in the fields of telecommunications and computer science, and has participated in several relevant national and international projects. He holds three U.S. patents.