Optimizing Access Demand for mMTC Traffic Using Neural Networks

Martí Llobet, Margarita Cabrera-Bean, Josep Vidal
Dept. of Signal Theory and Communications
Universitat Politècnica de Catalunya
Barcelona, Spain
{marti.llobet, marga.cabrera, josep.vidal}@upc.edu

Adrian Agustin
Centre Tecnològic de Telecomunicacions
de Catalunya (CTTC/iCERCA)
Castelldefels, Spain
adrian.agustin@cttc.cat

***Abstract*—Machine-type communications show unique spatial and temporal correlation properties that often lead to bursty access demand profiles. With the expected large-scale deployment of the Internet of Things (IoT), next-generation mobile networks should be redesigned to manage massive, highly synchronized arrivals of access requests by employing efficient access barring schemes. In this work, we first derived the analytical expression of the optimal Access Class Barring (ACB) parameter as standardized by the Third Generation Partnership Project (3GPP). Secondly, we predict the type and number of accessing devices from measurements acquired by the Base Station (BS) by employing Neural Networks (NNs). These estimates are used to effectively implement the optimal barring scheme, achieving performance results close to the theoretical bound.**

***Index Terms*—Neural Networks, Multi-Access Systems, Machine Type Communications**

I. Introduction

Unlike previous generations, 5G and beyond networks are not only focused on improving the system's broadband capabilities, but on enabling new use cases, notably massive Machine Type Communications (mMTC) and Ultra-Reliable Low-Latency Communications (URLLC). The massive deployment of low-complexity devices collecting and transmitting small amounts of data will redefine future mobile networks. Examples of MTC devices (MTCDs) include sensors for smart metering, and Internet of Things (IoT) and Internet of Everything (IoE) nodes, among other emerging technologies. With 75 billion IoT devices expected by 2025, mMTC will be a key enabler in upcoming 6G systems [1].

MTC traffic is based on the automatic exchange of sporadic short data packets, showing much higher spatial and temporal correlation compared to human-type traffic. However, the most critical type of traffic is generated by event-driven communications in highly condensed networks. For example, a typical household smart metering system will consist of a series of sensors measuring remarkably correlated features and reporting at very similar time intervals. Also, they will switch back and forth from connected to sleep mode to reduce power consumption, so that the number of connected devices will change frequently. Combining this situation with the presence of a *massive* amount of MTCDs, the signal capabilities of existing network access procedures will be usually exceeded. This scenario will require implementing an access control mechanism to encompass mMTC-specific use cases [2].

Access Class Barring (ACB) is the main access control mechanism for the Contention Based Random Access (CBRA) procedure, reducing the number of simultaneous access requests and redistributing them over time. Before initiating the network attachment procedure, accessing MTCDs are required to pass the barring check. The base station (BS) broadcasts the *barring parameter* and accessing MTCDs generate a random number; if it is larger than the barring parameter, the devices delay their access requests according to a *barring time*. Selecting an adequate barring parameter is not straightforward—it depends on the type and number of accessing MTCDs per time unit, maximizing the overall network throughput [3].

Selecting an adequate value for the barring parameter has been a recurrent research topic in the past years. The analytical optimal barring scheme derived in [4] has been presented as a departing point in similar, subsequent works tackling the issue. However, the authors did not fully adhere to the ACB protocol specified by the Third Generation Partnership Project (3GPP). These same authors proposed in [5] to dynamically allocate the number of available preambles depending on current traffic. The extensive analysis on the analytical expression of key performance indicators (KPIs) in congested mMTC environments presented in [6] is a remarkable work to consider. In [3], an in-depth analysis on the benefits of static barring schemes and their impact on KPIs was presented. In [7], ACB is introduced as a standard solution to alleviate congestion. Nevertheless, a uniform traffic profile is considered only, omitting highly synchronous activation patterns, a critical scenario in mMTC. A more recent work, [8], simplified the model to make it more tractable, describing it with a system of differential equations.

The latest works in the literature have concentrated on model-free schemes using Reinforcement Learning (RL). In [9], a table-based Q-learning (QL) agent employing discretized states selects the barring parameter from a set of 16 possi-

This work was partially supported by the project ROUTE56 (Agencia Estatal de Investigación, PID2019-104945GBI00/AEI/10.13039/501100011033), by the grant 2021 SGR 01033 (AGAUR, Generalitat de Catalunya) and by the European Union's Horizon Europe research and innovation programme through the project PREDICT-6G under grant agreement Nº 101095890.

The work of A. Agustin is supported by the Spanish Government through the project 6G AI-native Air Interface (6G-AINA, PID2021-128373OB-I00 funded by MCIN/AEI/10.13039/501100011033) and by "ERDF A way of making Europe"

ble values. Other authors have used *deep*-based approaches, meaning that the state-action pair is estimated using Neural Networks (NNs). In [10], a multiple-agent Deep QL (DQL) approach is proposed, considering three types of traffic characterized by different QoS requirements. In [11], spatial beamforming is introduced together with a two-class ACB scheme depending on the device's priority, generating an additional layer of complexity in the DQL environment. Other works use DRL-based approaches to tackle the issue of congested mMTC environments from a different perspective, such as in [12], where the authors focus on the optimal transmitting power and subchannel assignation for URLLC-type devices. In [13], a priority-dependent, group-based preamble reservation scheme is presented. Similarly to our approach, the cutting-edge [14] proposes employing Deep NNs (DNNs) as a means to estimate the optimal barring parameter. Their proposal uses a novel, non-standardized double-contention protocol, employing a DNN to infer the number of preamble collisions by processing the received signals.

The contributions of this work are the following:

1) A new analytical expression for the optimal barring parameter is presented. The new expression was derived by strictly adhering to 3GPP specifications, which state that the ACB scheme must only affect the MTCDs that have not yet transmitted their first preamble. To the best of our knowledge, no previous works deriving an optimal barring parameter met this requirement.
2) We propose an efficient NN-based algorithm that is capable of achieving the optimal bound in a real setting by exploiting our novel analytical expression of the optimal barring parameter. By contrast, the latest model-free RL-based approaches may lead to sub-optimal solutions that are computationally highly inefficient.

In short, the aforementioned contributions allow us to break down the problem of finding the optimal barring parameter into two different parts. First, by obtaining accurate estimates on the unknown level of network access overload using NNs and, secondly, by plugging these estimates into our novel analytical expression. When combined, our work's contributions provide a model-based approach allowing to apply the optimal ACB scheme in a practical setting. Our NN-based model remarkably differs from most state-of-the-art works, which employ model-free, RL-powered schemes that directly attempt to estimate the optimal barring parameter.

The paper is organized as follows: in Section II, the CBRA procedure and the ACB mechanism are presented; in Section III, our analytical expression for the optimal barring parameter is laid out; in Section IV, our NN-based model providing estimates of the optimal barring parameter is presented and tested; in Section V, performance results are discussed and compared with two state-of-the-art QL-based approaches. Finally, in Section VI, conclusions are drawn.

II. Access Class Barring in mMTC Applications

The 3GPP have specified two Random Access (RA) procedures for 5G New Radio (NR) systems: *Contention Based* RA (CBRA) and *Contention Free* RA (CFRA). The CBRA is the standardized procedure for first-time network attachments or re-attachment after a sleep period, whereas the CFRA is used when major modifications are made on ongoing connections [2]. As MTCDs are characterized by frequent, short-lived connections where few data packets are transmitted, the CBRA is the only RA procedure that is relevant to MTCDs.

The CBRA requires MTCDs to transmit their attachment requests in time-slotted channels known as Random Access Opportunities (RAOs). The whole procedure consists of a four-message handshake between accessing devices and the BS. Denoted *MSG1*, the first message is sent by the device and must contain a preamble selected at random from a pool of precomputed Zadoff-Chu sequences. It is the most critical CBRA step—if two or more devices request access using the same preamble, a collision occurs and the attachment attempt is declared as a failure for all the devices that had selected the same preamble. These devices will re-attempt network access after a random *backoff time* t_{BO}.

The 3GPP specify two different access demand profiles for MTC applications, denoted as *Traffic models 1* and *2* [15]. They are characterized by a uniform and a burst-like probability distribution, respectively, as to the amount of newly generated access requests per RAO. We concentrate on *Traffic model 2*, the most challenging scenario in mMTC. In this model, MTCDs access the network within the time frame $0 \leq t \leq T$, following a *Beta* distribution $g(t)$,

$$g(t) = \frac{t^{(\alpha-1)}\left(T-t\right)^{\beta-1}}{T^{(\alpha+\beta-1)}\mathbf{B}\left(\alpha,\beta\right)}\,, \tag{1}$$

with parameters $\{\alpha = 3, \beta = 4\}$, and $T = 10$ s. The $\mathbf{B}\left(\cdot\right)$ symbol designates the *Beta function*.

With the goal of maximizing the number of successfully connected MTCDs in scenarios with high access demand, ACB is proposed as the main access control mechanism by the 3GPP. In this scheme, the BS broadcasts the barring parameter $p \in [0, 1]$ together with other critical configuration parameters in the System Information Blocks (SIBs), which must be acquired by all accessing MTCDs with T_{SIB2} periodicity. Before initiating the CBRA procedure, all first-time access requests and those MTCDs that did not pass the ACB check in previous RAOs are subject to the ACB check. ACB-subjected devices generate a random number $q \sim \mathcal{U}\left[0, 1\right]$—if $q \leq p$, then the ACB check is passed. If that is the case, ACB-subjected devices are allowed to initiate the CBRA procedure. In the opposite case, ACB-subjected devices are applied a delay t_{barred}. MTCDs will terminate attempting to either pass the ACB check or initiate the CBRA procedure if the maximum number of per-device network access attempts, N^{max}, is reached. It must be obeyed that $N_u^c + N_u^d \leq N^{max}$, where N_u^c is the number of recorded preamble collisions by the device u. Similarly, N_u^d is the recorded number of failed ACB checks.

III. Deriving the optimal barring parameter

In this section, we present the derivation of a new analytical expression for the optimal barring parameter, p^*, following the 3GPP protocol. In contrast to previous works, we propose differentiating between two types of access requests to fully adhere to the 3GPP-specified ACB and CBRA protocols. The 3GPP standard states that those collided MTCDs that already passed the ACB check are not subject to it in future attachment attempts. This standard has the physical implication of increasing the collision probability in future RAOs, causing longer delays and higher energy consumption, and even being barred of attempting network access after exceeding the maximum number of access attempts N^{max}.

With the goal of counteracting the above mentioned negative effects, we propose classifying access requests into two groups: at the k-th RAO, $N_1\left[k\right] = n_1$ MTCDs are subject to the ACB check and must complete the CBRA stage to successfully attach to the network. Additionally, $N_2\left[k\right] = n_2$ previously collided MTCDs are not subject to the ACB check again, and directly attempt to complete the CBRA. Note that the total number of access requests at the k-th RAO is $N\left[k\right] = N_1\left[k\right] + N_2\left[k\right]$. Setting M as the total number of available preambles, the preamble m is selected by one out of the n_1 MTCDs, with a probability of p/M, where p is both the barring parameter and the probability of passing the ACB check. Similarly, a preamble m is selected by one out of the n_2 MTCDs, with a probability of $1/M$. Considering the two types of accessing devices, the preamble m will be selected by one single user with probability

$$\begin{aligned}&\Pr\{U_m\left[k\right]=1|N_1\left[k\right]=n_1,N_2\left[k\right]=n_2\}\\&=\binom{n_1}{1}\frac{p}{M}\left(1-\frac{p}{M}\right)^{n_1-1}\left(1-\frac{1}{M}\right)^{n_2}+\\&\left(1-\frac{p}{M}\right)^{n_1}\binom{n_2}{1}\frac{1}{M}\left(1-\frac{1}{M}\right)^{n_2-1}\end{aligned}, \tag{2}$$

where $U_m\left[k\right]$ denotes the number of mutually independent devices selecting preamble m at the k-th RAO. Note that in (2) the first term denotes the case where the MTCD that chooses preamble m is one of the n_1 devices attempting to pass the barring check. The second term denotes the case where the connected MTCD is one of the n_2 devices that experienced a preamble collision in previous RAOs. The expected number of successful transmissions at the k-th RAO, $N^s[k]$, conditioned to $N_1[k] = n_1$ and $N_2[k] = n_2$, can be computed as

$$\begin{aligned}&\mathbb{E}\{N^s[k]|N_1[k]=n_1,N_2[k]=n_2\}\\&=\sum_{m=1}^{M}\Pr\{U_m[k]=1|N_1[k]=n_1,N_2[k]=n_2\}\\&=n_1p\left(1-\frac{p}{M}\right)^{n_1-1}\left(1-\frac{1}{M}\right)^{n_2}+\\&\left(1-\frac{p}{M}\right)^{n_1}n_2\left(1-\frac{1}{M}\right)^{n_2-1}\end{aligned}. \tag{3}$$

This above expression is critical for obtaining the optimal barring parameter p^*. Derivating and solving for p, p^* is the one maximizing (3), with the following expression

$$p^* = \max\{0, \min\{1, \frac{M(M-1-n_2)}{n_1(M-1)-n_2}\}\}. \tag{4}$$

With the purpose of providing a more in-depth analysis on the goodness of our theoretical derivations, we also include the expressions of other critical features. (5) is the average number of preambles that are not selected by any MTCD at the k-th RAO, $N^0[k]$, with the following expression

$$\begin{aligned}&\mathbb{E}\{N^0[k]|N_1[k]=n_1,N_2[k]=n_2\}\\&=\sum_{m=1}^{M}\Pr\{U_m[k]=0|N_1[k]=n_1,N_2[k]=n_2\}\\&=M\left(1-\frac{p}{M}\right)^{n_1}\left(1-\frac{1}{M}\right)^{n_2}\end{aligned}. \tag{5}$$

Then, the average number of preambles experiencing a collision at the k-th RAO, $N^c[k] = M - N^s[k] - N^0[k]$, is obtained by combining (3) and (5), as follows

$$\begin{aligned}&\mathbb{E}\{N^c[k]|N_1[k]=n_1,N_2[k]=n_2\}\\&=M-\left(1-\frac{p}{M}\right)^{n_1-1}\left(1-\frac{1}{M}\right)^{n_2-1}\\&\left(M+p(n_1-1)+(n_2-1)-\frac{p}{M}(n_1+n_2-1)\right)\end{aligned}. \tag{6}$$

IV. NN-BASED PREDICTION OF ACCESS REQUESTS

As the number of access requests $(N_1[k], N_2[k])$ are not available data at the base station, we propose predicting these features by employing NNs that infer their value from other available measurements associated with the level of network access overloading. Then, the NN-obtained estimates $(\hat{N_1}[k], \hat{N_2}[k])$ may be plugged into (4) to obtain the next barring parameter update. As the barring parameter is updated every T_{SIB2} period (which will obey $T_{SIB2} = nT_{RAO}$, $n > 0$), the per-RAO averages over the l-th SIB2 period, $\left(\overline{N}_1[l], \overline{N}_2[l]\right)$, are used instead. These averages follow the expression, $\overline{N}_i[l] = \frac{T_{RAO}}{T_{SIB2}} \sum_{k=1}^{T_{SIB2}/T_{RAO}} N_i[k]$, for $i = 1, 2$, where T_{RAO} is the RAO periodicity. This same notation is employed for the rest of features.

The network input vector at the l-th SIB2 period is $\boldsymbol{x}[l] = \{\overline{N}^s_1[l-1], \overline{N}^s_2[l-1], \overline{N}^c[l-1], \overline{D}_{avg}[l-1], p[l-1]\}$, with $l > 0$ and $\boldsymbol{x}[0] = \{0, 0, 0, 0, 1\}$. N^s_1 designates the number of MTCDs that successfully completed the CBRA procedure and never experienced a collision; N^s_2 designates those devices that also successfully completed the CBRA procedure, but experienced one or more collisions in previous RAOs; N^c is the number of collided preambles detected by the BS; D_{avg} is the per-device mean sum of accumulated t_{BO} and t_{barred} delays of connected MTCDs, measured in RAOs; and $p \in [0, 1]$ is the barring parameter. The network output at the l-th SIB2 period are the NN-obtained estimates, contained in vector $\hat{\boldsymbol{y}}[l] = \{\hat{N}_1[l], \hat{N}_2[l]\}$. The input and output layers are connected through several intermediate *fully connected* layers, with the output at the j-th layer, $\boldsymbol{r}^{(j)}$, given by

$$\boldsymbol{r}^{(j)} = f\left(\boldsymbol{w}^{(j)}\boldsymbol{s}^{(j-1)} + \boldsymbol{b}^{(j)}\right), \tag{7}$$

where f is the activation function, $\boldsymbol{s}^{(j-1)}$ is the input at the j-th layer, $\boldsymbol{w}^{(j)}$ is the weight matrix between layers $j-1$ and j, and $\boldsymbol{b}^{(j)}$ is the bias applied on layer j's output. The activation function f is a Rectified Linear Unit (ReLu). In Fig. 1, a graphic schematic summing up the two-step, NN-based approach described in this section is presented.

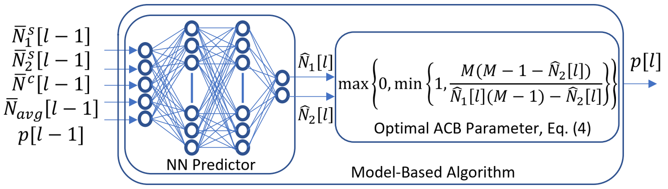

Fig. 1. Graphic schematic on our proposed two-step, NN-based approach to estimate the optimal ACB parameter.

The values of the weights and biases are obtained iteratively according to the stochastic gradient descent with momentum (SGDM) optimizer. During the training phase, the network uses the response feature vector $\bar{\boldsymbol{y}}[l] = \{\overline{N}_1[l], \overline{N}_2[l]\}$ to compute the loss $\mathcal{L}$, which is set to be the Mean Square Error (MSE), with the expression $\mathcal{L} = \frac{1}{2}\|\bar{\boldsymbol{y}}[l] - \hat{\boldsymbol{y}}[l]\|^2$.

The databases used to train the NNs were generated using concatenated *episodes*. We define each *episode* as the time frame $[0, T_{epi}]$ during which measurements are recorded. With the goal of capturing long-delayed devices, we set $T_{epi} = 2T$, where T defines the time frame during which $U = 30,000$ new access requests arrive following the $Beta(3,4)$ distribution as in (1). Four databases were generated, each characterized by the unique combination of the following features' values, $N^{max} = \{5, 10\}$ and $T_{SIB2} = \{1, 16\}T_{RAO}$, as shown in Table I. In all databases, the barring parameter is updated according to the optimal policy given by (4). Each database was partitioned into three data sets—700 episodes in the training set, and 300 episodes in the validation and test sets. The training database is applied a *z-score* normalization before being passed on to the NN. The obtained z-score parameters—the mean μ and the deviation σ—are applied on both the validation and test databases. In all databases, each RAO has a duration of 5 ms and the number of available preambles is set at $M = 54$. The delay applied on the devices that do not pass the ACB check is computed as $t_{barred} = (0.7 + 0.6\,\mathcal{U}\,[0,1])\,T_{acb}$, with $T_{acb} = 2^{N_u^d}$. Similarly, the delay applied on collided devices is set at $t_{BO} \sim \mathcal{U}\,[0, T_{BO}]$, with $T_{BO} = 2^{N_u^c}$.

TABLE I
LIST OF SELECTED NNS

N^{max}	**Train** T_{SIB2}	**Layer Struc.**	**Test RMSE for** N_1	**Test RMSE for** N_2
5	T_{RAO}	5-10-2	1.769	1.363
5	$16\,T_{RAO}$	5-10-10-2	0.599	0.448
10	T_{RAO}	5-10-2	1.780	1.370
10	$16\,T_{RAO}$	5-10-10-2	0.607	0.451

For each database listed in Table I, different layer structures (in terms of the number of hidden layers and number of neurons) were evaluated. The best-performing NNs in terms of the RMSE on the test databases were selected. In all cases, the training was performed on 30 epochs (the number of full passes on the entire training database). Note that lower RMSE values are obtained for $T_{SIB2} = 16\,T_{RAO}$ as recorded measurements are averaged over several RAOs. This positively impacts on the estimation error due to the lower variance values.

V. Results

In this section, we present and discuss the evolution of several KPIs to assess the goodness of our proposed barring scheme. All simulation results presented in this section share the following common parameters. Specifically, the number of available preambles is set at $M = 54$; the time frame in which new access requests are generated, is set at $T = 10$ s; the time frame T_{epi} during which measurements are recorded is used as a tunable hyperparameter, and is assigned the value that provides the best results in each simulated approach; the number of MTCDs attempting networks access during an episode according to *Traffic model 2* is set at $U = 30,000$, the maximum standardized value [15]; and the duration of a Random Access Opportunity (RAO) is set at $T_{RAO} = 5$ ms. Two other parameters may take two different values depending on the simulation setting: the transmission periodicity of SIB2 blocks is set at $T_{SIB2} = \{1, 16\}T_{RAO}$, and the maximum number of per-device network access attempts is set at $N^{max} = \{5, 10\}$. Also, the time evolution of the analyzed KPIs is presented in terms of RAOs instead of SIB2 blocks for the sake of comprehensiveness.

In Fig. 2, some preliminary results assessing the goodness of our derived analytical expressions are presented. The analytically obtained parameters are compared against their counterpart simulated values. We generate 10^3 extended episodes, updating the barring parameter according to (4) with $T_{SIB2} = T_{RAO}$ periodicity. The per-RAO average number of served devices, $\mathbb{E}\{N^s[k]\}$, and the per-RAO average number of collisions, $\mathbb{E}\{N^c[k]\}$, are computed following two different methods. The so-called *analytical* method consists in using equations (3) and (6). These equations depend on the number of recorded access requests, (n_1, n_2), which are modeled as random variables. We thus employ their averages, $\mathbb{E}\{n_i[k]\}$, over the 10^3 simulation runs and compute the parameters using these values. The other method, named *simulated*, consists in directly measuring $(N^s[k], N^c[k])$ in each simulation run and subsequently providing their averaged values. It can be observed how the analytical method closely follows the simulated one, showing remarkably similar values.

In Table I, we listed the four validated NNs used to predict the number of access requests (N_1, N_2), one for each considered pair of possible training values assigned to the parameters N^{max} and T_{SIB2}. In the simulation results, the policies obtained through NNs that were trained using $T_{SIB2} = \{1, 16\}T_{RAO}$ are assigned case numbers *2* and *3*, respectively, leaving Case *1* for the analytical optimal policy. Both NNs were trained using a gradient decay factor of 0.9, and learning rate of 10^{-3}. In the simulation environment used to obtain Table III and Fig. 3, the barring parameter p is updated every $T_{SIB2} = 16\,T_{RAO}$.

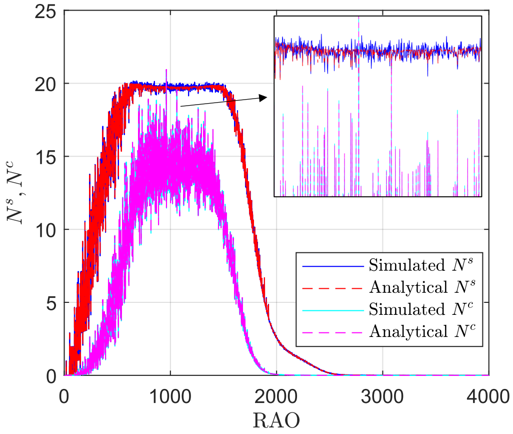

Fig. 2. RAO-by-RAO evolution of the average number of served MTCDs, $N^s[k]$, and reported preamble collisions, $N^c[k]$, over 10^3 runs of an episode using the optimal barring scheme provided by (4). *Simulated* results are average recorded measurements, whereas the *analytical* ones are computed using (3) and (6) through the average recorded measurements of $p[k]$, $N_1[k]$ and $N_2[k]$. Simulation parameters: $T_{SIB2} = T_{RAO}$; $N^{max} = 10$; and $T_{epi} = 2T$

Designated Cases *4* and *5*, we included the barring policies acquired by a DQL agent and a table-based QL agent, respectively, as additional state-of-the-art benchmarks. Both techniques are model-free RL-based algorithms that learn to map a value known as *reward*—in this case, the cumulative expected number of served devices, $\mathcal{R} = \sum_l \overline{N}^s[l]$—to an *action* a—in this case, the discretized ACB parameter, $a = p^{QL} \in \{0.05, 0.1, \ldots, 1\}$—given a particular state vector $\boldsymbol{s}^{QL}$ containing relevant measurements of the environment. In this simulation setting, the selected state vector is $\boldsymbol{s}^{QL}[l] = \{N^s[l-1], N^c[l-1], D_{avg}[l-1], p[l-1]\}$ (where $N^s[l-1] = N_1^s[l-1] + N_2^s[l-1]$), with $l > 0$, $\boldsymbol{s}^{QL}[0] = \{0, 0, 0, 1\}$. The reward is estimated by the function $Q\left(\boldsymbol{s}^{QL}, p\right)$, also known as *critic*, which corresponds to a DNN or a table for a DQL or discretized QL agent, respectively. The corresponding action a—the next barring parameter update—is then obtained as $p[l] = arg\max_a Q\left(\boldsymbol{s}^{QL}[l], a\right)$. Both Cases 4 and 5 were trained with an update periodicity of the barring parameter of $T_{SIB2} = T_{RAO}$ and an episode duration of $T_{epi} = 1.5\,T_{RAO}$. Also, the two RL agents were trained *offline* for 10^4 episodes, keeping the policy learnt in the episode showing the highest reward $\mathcal{R}$. The simulation environment used to test and present the performance results of the acquired policies applies an update periodicity of the ACB parameter of $T_{SIB2} = 16\,T_{RAO}$, as indicated in 3GPP standards. The following simulation settings, however, are unique to Cases 4 and 5, respectively:

1) The DQL agent has been designed to resemble as much as possible to the single-agent policy presented in [10]. The employed DNN is *feed forward* and features a 10-neuron fully connected hidden layer, with its weights being updated every $N_B = 32$ RAOs. Also, the discount factor is set at $\gamma = 0.9$.
2) The table-based QL agent has been designed to follow the approach presented in [9] as closely as possible. This uses discretized values for all state features, quantifying them in 10-value intervals. The only exception is D_{avg}, which is quantified in 3 intervals.

All simulated ACB policies and their case numbers are listed in Table II. The *uniform* policy—p always set to 1—is also included in the simulation results as a baseline validation.

TABLE II
LIST OF SIMULATED ACB POLICIES

Case No.	ACB Policy	State Space	Train T_{SIB2}	Train T_{epi}
(1)	Optimal, analytical	—	—	—
(2)	Optimal, NN-estimated	$\boldsymbol{s}$	T_{RAO}	$2T$
(3)	Optimal, NN-estimated	$\boldsymbol{s}$	$16\,T_{RAO}$	$2T$
(4)	DQL agent	$\boldsymbol{s}^{QL}$	T_{RAO}	$1.5T$
(5)	Table-based, QL agent	$\boldsymbol{s}^{QL}$	T_{RAO}	$1.5T$

In Fig. 3, the RAO-by-RAO evolution of the estimated barring parameter $p[k]$ and four KPIs throughout an episode are depicted for the ACB policies listed in Table II and the uniform policy, with $N^{max} = 10$. The featured KPIs are the number of served MTCDs $N^s[k]$, the number of reported collisions $N^c[k]$, the per-device average accumulated delay of connected MTCDs $D_{avg}[k]$, and the Cumulative Distribution Function (CDF) of the accumulated delays reported by connected MTCDs. The obtained results show that the NN-based approaches (Cases *2* and *3*) and the DQL agent (*Case 4*) closely follow the analytical bound given by *Case 1*, remarkably outperforming the uniform policy. While Cases 1–4 show similar $N^s[k]$ values during the entire episode, they differ as to the evolution of the number of collisions $N^c[k]$. The model-based policies (Cases 1–3) and the DQL agent show a maximum value of $N^c[k]$ that is around 46% and 36% lower, respectively, than that of the uniform policy. However, their main difference is the fact that the optimal-based policies manage to keep $N^c[k]$ values uniform throughout the peak of access demand, whereas the DQL agent fails to do that. Also, the optimal-based policies manage to obtain lower D_{avg} values than the DQL agents. The table-based QL agent (Case 5) lags considerably behind Cases 1–4, though it does outperform the uniform policy.

To further evaluate our NN-based barring scheme, in Table III, we list average final KPI values for the cases $N^{max} = \{5, 10\}$, featuring the ACB policies of Table II and the uniform policy. The KPIs featured in the table are the rate of successfully attached MTCDs, P_s; the per-device total accumulated delay of connected MTCDs, D_{avg}; and the delay 95-th percentile, D_{95}, that is the per-device maximum observed accumulated delay in RAOs shown by 95% of connected MTCDs. As indicated in Table III, Cases 1–4 show statistically identical P_s values, but differ on the rest of final KPIs. As expected KPI values for Cases 1–3 are almost identical, with Case 3 showing a lower D_{95} than the analytical case. This result is an indication that using averaged results over a 16-RAO window generates a smoother barring policy that better adapts to the measurements' noisy profiles.

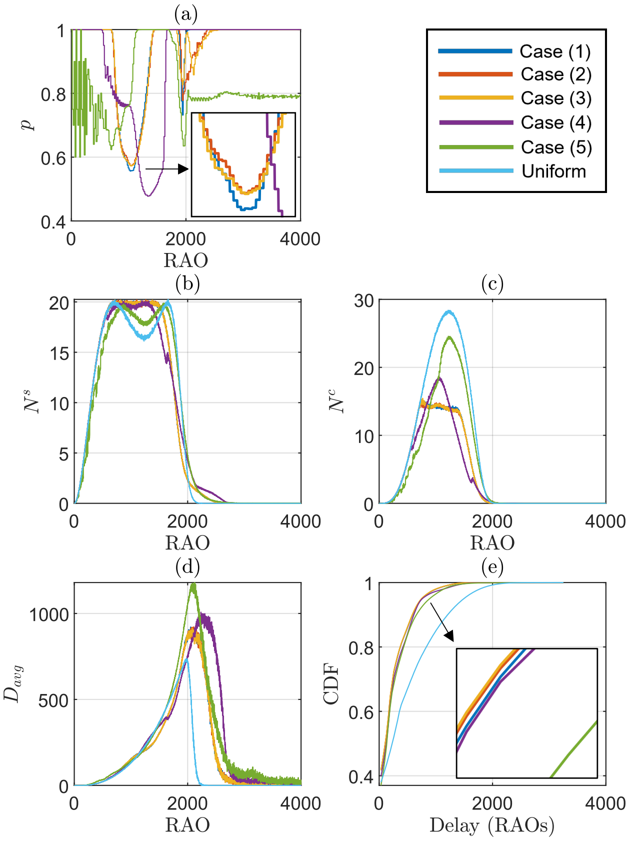

Fig. 3. RAO-by-RAO evolution averaged over 10^3 simulation runs, with $N^{max} = 10$ and $T_{epi} = 2T$, of: (a) the barring parameter $p[k]$; (b) the no. of served MTCDs $N^s[k]$; (c) the no. of preamble collisions $N^c[k]$; (d) the per-device accumulated delay of connected MTCDs $D_{avg}[k]$; and (e) the Cumulative Distribution Function (CDF) of the accumulated delay of connected MTCDs. Results shown for the Table II cases and the *uniform* policy.

TABLE III
AVGD. FINAL KPI VALUES FOR THE ACB POLICIES IN TABLE II

Case →	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**Unif.**
$N^{max} = 5$						
P_s	0.9835	0.9834	0.9836	0.9843	0.962	0.927
D_{avg}	187.3	187.4	187.7	210.0	241.9	189.8
D_{95}	690.9	693	686.7	699.3	945.0	568.4
$N^{max} = 10$						
P_s	0.9996	0.9996	0.9996	0.9997	0.9984	0.9938
D_{avg}	201.94	201.95	201.96	223.49	271.88	228.89
D_{95}	747.5	752.1	721.6	761.3	1017.9	787.5

VI. CONCLUSIONS

In this work, we presented a novel analytical expression for the optimal ACB parameter that adheres to 3GPP specifications. Besides, a NN-based model to obtain accurate predictions on the type and number of access requests in congested mMTC environments was successfully tested, enabling the BS to exploit these additional data to apply the optimal barring parameter using available measurements. The simulation results assess the goodness of our proposed approach, which closely follows the analytical bound—it manages to achieve the same number of successfully connected MTCDs and keeps a stable number of collisions during the peak of access demand, just like under optimal conditions. We also show that our NN-based approach outperforms model-free DQL agents operating under the same conditions.

REFERENCES

[1] N.H. Mahmood et al. "Six Key Features of Machine Type Communication in 6G". In: *2nd 6G Wireless Summit*. 2020, pp. 1–5.

[2] I. Leyva-Mayorga et al. "Random Access for Machine-Type Communications". In: *Wiley 5G Ref: The Essential 5G Reference Online* (2019), pp. 1–21.

[3] L. Tello-Oquendo et al. "Performance Analysis and Optimal Access Class Barring Parameter Configuration in LTE-A Networks With Massive M2M Traffic". In: *IEEE Trans. on Vehicular Technology* 67.4 (2018), pp. 3505–3520.

[4] S. Duan, V. Shah-Mansouri, and V.W.S. Wong. "Dynamic access class barring for M2M communications in LTE networks". In: *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013, pp. 4747–4752.

[5] S. Duan et al. "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks". In: *IEEE Transactions on Vehicular Technology* 65.12 (2016), pp. 9847–9861.

[6] C. Wei, G. Bianchi, and R. Cheng. "Modeling and Analysis of Random Access Channels With Bursty Arrivals in OFDMA Wireless Networks". In: *IEEE Trans. on Wireless Communications* 14.4 (2015), pp. 1940–1953.

[7] V. Shah-Mansouri, S. Srinivasan, and V.E. Balas. "Optimal Access Class Barring in Machine to Machine Systems with Random Activation Time". In: *Informatica* 28.2 (2017), pp. 285–302. ISSN: 0868-4952.

[8] Y. Hadjadj-Aoul and S. Ait-Chellouche. "Access Control in NB-IoT Networks: A Deep Reinforcement Learning Strategy". In: *Information* 11.11 (2020). ISSN: 2078-2489.

[9] L. Tello-Oquendo et al. "Reinforcement Learning-Based ACB in LTE-A Networks for Handling Massive M2M and H2H Communications". In: *IEEE Int. Conference on Communications*. 2018, pp. 1–7.

[10] Z. Chen and D.B. Smith. "Heterogeneous Machine-Type Communications in Cellular Networks: Random Access Optimization by Deep Reinforcement Learning". In: *2018 IEEE Int. Conference on Communications*. 2018, pp. 1–6.

[11] A. Agustin, J. Vidal, and M. Cabrera-Bean. "Hierarchical Beamforming in Random Access Channels". In: *IEEE Global Communications Conference*. 2021, pp. 1–6.

[12] H. Yang et al. "Deep Reinforcement Learning Based Massive Access Management for Ultra-Reliable Low-Latency Communications". In: *IEEE Trans. on Wireless Communications* 20.5 (2021), pp. 2977–2990.

[13] T.N. Weerasinghe, I.A.M. Balapuwaduge, and F.Y. Li. "Preamble Reservation Based Access for Grouped mMTC Devices with URLLC Requirements". In: *IEEE Int. Conference on Communications*. 2019, pp. 1–6.

[14] C. Zhang et al. "Deep Learning Based Double-Contention Random Access for Massive Machine-Type Communications". In: *IEEE Transactions on Wireless Communications* (2022), pp. 1–14.

[15] 3GPP. *Study on RAN Improvements for Machine-Type Communications (Release 11)*. Technical Specification (TS) 37.868. 3rd Generation Partnership Project (3GPP), 2011.