Adjustable Access Control Mechanism in Cellular MTC Networks: A Double Q-Learning Approach

Diego Pacheco-Paramo*, **Luis Tello-Oquendo**†

*Escuela de Ciencias Exactas e Ingeniería, *Universidad Sergio Arboleda*, Bogotá 111221, Colombia
†College of Engineering, *Universidad Nacional de Chimborazo*, Riobamba 060108, Ecuador

***Abstract*—The potential of a broad set of applications that might impact different economic sectors rely on their ability to capture, process, and analyze large amounts of data accurately. Concerning wireless networks, one of the main issues that arise in this type of scenarios is access control. In Machine Type Communication scenarios, a vast number of devices could access the network simultaneously, which can harness other types of users. A well-known congestion control method used in cellular networks is access class barring (ACB); it has been designed to allow a large number of users to access the network even under high traffic conditions by delaying their access attempts. However, its performance has to adapt dynamically to traffic conditions while at the same time, providing enough flexibility to administrators to prioritize crucial key performance indicators. We propose a Double Q-learning-based ACB mechanism that adapts dynamically to different traffic conditions when machine-to-machine and human-to-human communications coexist, with three different configurations that allow to adjust the trade-off between successful access probability and mean access delay. The results show that each configuration can reach different levels of QoS related to successful access probability, mean access delay, and mean number of preamble transmissions.**

***Index Terms*—double Q-learning; Internet of Things; access control; 5G; machine type communications; access class barring.**

I. Introduction

Internet of Things (IoT), the network of interconnected physical objects, is rising as a disruptive technology that concedes to the things the ability to sense, communicate, interact, and collaborate in such a way society evolves in several aspects with practical applications. Nowadays, the IoT concept has gained enormous momentum; at this rapid rate of growth, it is expected a huge number of interconnected devices, nearly 29 billion by 2022 [1], deployed in several applications and increasing the global mobile data traffic to 49 exabytes (10^{18} bytes) by 2021 [2]. The preeminent facilitator of the IoT ecosystem is Machine-type communication (MTC), also known as machine-to-machine (M2M) communication, which enables ubiquitous applications and services.

Several peculiarities of M2M traffic require specialized and inter-operable communication technologies that can both satisfy the highly-demand QoS requirements and handle a significant part of this emerging traffic. To this end, cellular networks are the most suitable choice due to their already deployed infrastructures, extensive coverage area, and high-performance capabilities [3], [4].

Dealing with the cumbersome number of connections and signaling produced when a vast number of user equipment (UEs) attempts to access the cellular network is an essential challenge that has gained a significant amount of attention in current research. Besides, another aspect to consider is the limited resources of the physical channels when UEs perform the random access procedure.

The access class barring (ACB) mechanism is one of the solutions included in the LTE-A radio resource control specification [5] to alleviate the issues mentioned above. It is a congestion control mechanism that regulates the concurrent access and protects the system's performance and service quality. In this sense, ACB expands the UE accesses through time randomly, aiming at delaying the arrivals of the UE access attempts in consonance with a barring rate and a barring time.

When the ACB mechanism is operating, its configuration parameters must adequately be adjusted since there is a trade-off between mitigating congestion and the performance of the network [6]. Furthermore, the proper tuning of the ACB parameters according to the traffic intensity is critical, but it is not a trivial task. For that reason, no directions have been specified by the 3GPP to implement the ACB mechanism effectively and to tune its configuration parameters in such a way they can dynamically and autonomously readjust to the network load. Bear in mind that the dynamic adaptation of the barring rate is beneficial since implementing a static ACB influence the access delay of every UE, even in circumstances when there is no congestion in the network, and the ACB mechanism is not required at all.

This paper aims to propose a double Q-learning algorithm to tune the ACB barring rate dynamically by considering both the traffic load in the random access channel (RACH) and its capacity. We implement three different configurations which allow providing different QoS for the network key performance indicators (KPIs) of interest. Our solution conforms with current system specifications and can be used to perform efficient congestion control and to facilitate the coexistence of H2H and M2M traffic.

The article proceeds as follows. Section II conducts a review of the relevant literature regarding ACB. Section III present the random access procedure in LTE-A and the ACB mechanism. Section IV provides a detailed explanation of the proposed double Q-learning approach to tune ACB jointly with its implementation. Section V presents the performance evaluation and our most relevant results. Finally, Section VI draws the conclusions.

II. Related Work

Numerous proposals aiming at optimizing the access control for handling massive MTC connection attempts on the RACH have been introduced through either static or dynamic approaches [7]–[9]; however, many of them are based on

complex procedures, do not comply with standards guidelines (e.g., the updating period of notification information by the base station is not considered), use strong assumptions for gaining high performance, and even misinterpret the operation of this mechanism [8], [10], [11].

Reinforcement learning has already been proposed to enhance the efficiency of access control in cellular MTC networks. In [12], we proposed an adaptive access control mechanism using Q-Learning when both M2M and H2H UEs coexist. In [13], a Q-learning approach is also proposed, but it considers a single type of traffic and assumes that the base station has full knowledge of the contending UEs. Also, in [14] a Q-Learning mechanism that prioritizes traffic (M2M and H2H) is proposed, although it is assumed that the base station has full knowledge of the number of contending UEs of each type. In [15], an access control mechanism is proposed based on Q-learning without ACB, where each UE learns when to transmit. However, this solution fails to meet the required QoS in different KPIs. In [16], we proposed a deep reinforcement learning mechanism to adapt the ACB mechanism to changing traffic conditions. Also, in [17], a deep reinforcement learning solution is proposed for dynamic access control. However, these solutions tend to rely heavily on the computational capacity of the base station or the nodes due to the training associated with neural networks. In this paper, we aim at evaluating the performance of a solution that does not require a substantial computational cost.

III. LTE-A Random Access Procedure and Access Control

Every time an MTC device requires to access the network, it must first acquire some configuration parameters from the base station. Among these parameters are the predefined time/frequency resources in which the random access attempts are allowed. Each occurrence of these resources, in which an access attempt can be made, is called a Random Access Opportunity (RAO); the sequence of RAOs constitutes what is called the RACH. The *Master Information Block* (MIB) and the *System Information Blocks* (SIBs) are resources used by the base station to broadcast the configuration information periodically. Once the UEs get this information, they perform a random access procedure using the RACH to signal the connection request [18], [19].

As detailed in [6], [18], [20]–[22], the random access consists in a four-message handshake contention-based procedure. In *Msg1*, a UE transmits a randomly chosen preamble from the preamble pool during one of the available RAOs. A preamble will be detected at the base station if it has not been chosen by more than one UE in the same RAO. Otherwise, a collision occurs. Then, the base station sends a random access response message, *Msg2*, which includes one uplink grant for each detected preamble. *Msg2* is used to assign time-frequency resources to the UEs for the transmission of *Msg3*. UEs wait for a predefined time window to receive the uplink grant. If no uplink grant is received by the end of this window and the maximum number of access attempts has not been reached, the UEs wait for a random time and then perform a new access attempt. That is, they select a new preamble and transmit it at the next RAO. The UEs that receive an uplink grant send their connection request message, *Msg3*, using the resources specified by the base station. Finally, the base station responds to each *Msg3* transmission with a contention resolution message, *Msg4*.

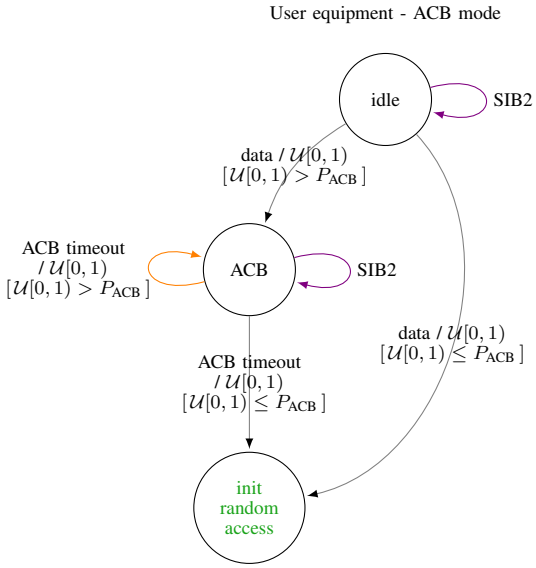

Figure 1. State transition diagram of the ACB mechanism when access control is implemented for M2M UEs. *Transition notation:* event / action [condition]

Access Class Barring (ACB) is a congestion control mechanism devised for restricting the number of concurrent access attempts from UEs. The main goal of ACB is to redistribute the access requests of UEs through time; by doing so, the number of access requests per RAO is reduced. This fact helps to evade massive-synchronized access demands to the RACH, which might endanger the fulfillment of QoS objectives. The state transition diagram of the ACB mechanism [5], [21] is depicted in Fig. 1. Note that ACB is applied only to the UEs that have not yet begun its random access procedure as explained above.

If ACB is not operating, all UEs are allowed to access the RACH. When ACB is operating, the base station broadcasts (through SIB2) both mean barring times, $T_{\text{ACB}} \in \{4, 8, 16, \ldots, 512\,\text{s}\}$, and barring rates, $P_{\text{ACB}} \in \{0.05, 0.1, \ldots, 0.3, 0.4, \ldots, 0.7, 0.75, 0.8, \ldots, 0.95\}$. Then, each UE should determine its barring status with the information provided from the base station (i.e., P_{ACB}, T_{ACB}) before starting the random access procedure. For doing so, the UE generates a random number between 0 and 1, $\mathcal{U}[0,1)$. If this number is less than or equal to P_{ACB}, the UE proceeds with its *Msg1* transmission. Otherwise, the UE waits for a random time calculated as follows

$$T_{barring} = [0.7 + 0.6\,\mathcal{U}[0,1)]\; T_{\text{ACB}}. \tag{1}$$

Note that ACB is only useful when a massive number of UEs attempt transmission at a given time but the system is not continuously congested.

IV. Implementing the Double Q-learning Mechanism

In this section, we define the Markov Decision Process (MDP) representing the interaction of the base station with UEs. Then, we describe our implementation of Double Q-Learning [23], a well-known reinforcement learning algorithm. This algorithm does not suffer from the overestimation of Q

values that might occur in traditional Q-learning [24], causing poor performance. The state space and action set that define our MDP was already seen in [12].

A. System Model

In LTE-A, the base station can know the number of UEs that are contending for resources based on the number of successfully received preambles, N_{psu}. That is, the number of correctly received preambles does not necessarily coincide with the number of sent preambles. Therefore, the base station only receives observations that provide some information about the real state of the system, or the number of UEs contending for resources. Based on N_{psu}, the base station should set a value for the barring rate P_{ACB}, hoping to reduce the congestion and thus allowing more UEs to access the network. However, it is expected that providing more information to the base station allows for better decision-making.

As explained in Section III, there are specific radio resources (or RAOs) for access contention, and the number of RAOs per frame is given by the *prach-ConfigIndex* parameter [19], [21]. Also, the method through which a base station can inform UEs about the new barring rate to be used is a SIB2 message. This message is sent every 80 ms, a longer period than that of the RAOs. In fact, in this configuration there are 16 RAOs between two SIB2 messages. Therefore, the setting of P_{ACB} will affect 16 RAOs.

Hence, a state has to consider what is happening in more than one RAO, and the traffic that affects two RAOs in a row might be very different. Therefore, we will consider the mean number of successfully decoded preambles between two SIB2 messages, $\overline{N_{psu}}$, as a first parameter of the state of the system. Since the number of N_{psu} might change in every RAO, we will use the variation coefficient of the measurements of N_{psu} between two SIB2 messages, $CV_{N_{psu}}$. Also, we are interested in knowing if there are changes on traffic, and therefore we include the value ΔN_{psu}, which is the difference of $\overline{N_{psu}}$ between the current period and the previous one, allowing us to understand how the traffic is changing. Finally, because the decision on P_{ACB} defines how many UEs can access the system, we also include this value on our state definition. Therefore, a state s, is defined as $s = \left(\overline{N_{psu}}, CV_{N_{psu}}, \Delta N_{psu}, P_{\text{ACB}}\right)$. Fig. 2 illustrates how the state is defined according to the SIB2 messages and N_{psu} for each RAO. One important aspect that we should recall in LTE-A is that although there are 54 available preambles, only 15 preambles can be successfully acknowledged in a RAO. Therefore, even if more than 15 preambles are successfully decoded in the base station, at least one UE is not going to be able to connect to the network. Remember that on each state, as defined earlier, the system has to decide what action is going to take to reduce the congestion, if at all. Therefore, it is evident that the action in our MDP is P_{ACB}. The values that P_{ACB} can take are $\{0.05, 0.1, 0.15, .., 0.3, .., 0.7, 0.75, .., 0.95, 1\}$; that is, we have 16 possible actions in our system. This characteristic, and the fact that many UEs can choose the same preamble causing collisions makes that when $\overline{N_{psu}} > 29$, the impact of P_{ACB} is marginal. Therefore, we aggregated all the states where this condition is fulfilled in the state where $\overline{N_{psu}} = 29$. Also, in order to reduce the number of states, we discretized $CV_{N_{psu}}$ such that $CV_{N_{psu}} \in \{0, 0.2, 0.4, 0.6, 0.8\}$. Finally, ΔN_{psu} only informs if there have been changes on $\overline{N_{psu}}$, not the amount of those changes.

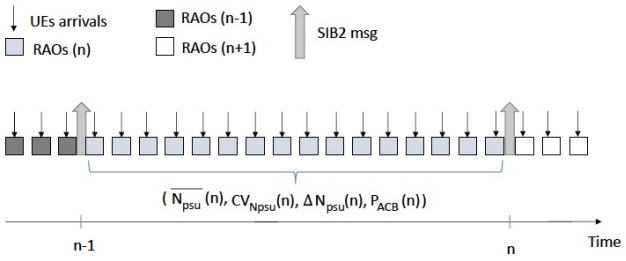

Figure 2. State definition with RAOs and SIB2.

B. Double Q-Learning Mechanism

In this section, we explain our implementation of Double Q-Learning. A Q-Learning [24] mechanism has already been implemented in [12] with the same state space and actions. However, one of the issues that might arise with Q-Learning is that it can overestimate the Q-values since the target uses the same estimator to obtain the action that maximizes the return at state s^0. This behaviour might end up rewarding state-action combinations beyond their real value, making that the algorithm does not converge to an adequate solution. In [23], the authors propose a method based on a double estimator that avoids this overestimation. This characteristic can help avoiding sub-optimal solutions. We implement Double Q-learning as presented in [23] by maintaining two different Q-tables, as it can be seen in the following expression:

$$Q(s,a)_1 = Q(s,a)_1 + \alpha \left[\mathcal{R} + \gamma \max_{a' 2 A} [Q(s^0, a^0)_2] - Q(s,a)_1\right], \quad (2)$$

the update of Q-table 1 uses the estimate from Q-table 2. The implementation is shown in Algorithm 1. In this case, the values of α, γ and $\mathcal{R}$ are the same as in the Q-learning algorithm shown earlier. In the case of ϵ, it is set to define when the greedy policy is used. Obviously, this approach requires more memory due to the use of two Q-tables, but in this case by the state definition shown earlier, the size of these tables is not too big.

C. Reward Function

A fundamental aspect that defines the behavior of the Double Q-learning algorithm is the reward function, which associates cost/rewards to the actions taken on states. Since we have a large number of states, it is impossible to set a specific cost/reward to each state/action combination. In order to establish some guidelines that allows to define this function, we have followed the approach proposed in [20] where the distribution of UEs contending for resources on a RAO is obtained as a function of the number of successfully decoded preambles in the base station. Also, in [20] the authors assert that having 20 UEs contending maximizes the probability of successfully receiving 15 UEs, which is the maximum number of preambles that the base station can acknowledge. However, this value also increases the probability of congestion. It might be of the interest for the network administrator to be able to adjust the performance of the system according to a

Algorithm 1: Double QL-Based ACB Mechanism

Controller: Double Q-learning($\mathcal{S}, \mathcal{A}, \alpha, \mathcal{R}, \gamma, \epsilon$)
Input : $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, α is the learning rate, $\mathcal{R}$ is the reward, γ is the discount factor, ϵ is the exploration probability
Local : real array $\mathbf{Q}[s,a]_A$, state s, action a
Local : real array $\mathbf{Q}[s,a]_B$, state s, action a

repeat
  if *RAO*$(i) \mod T_{SIB2} = 0$ **then**
    select action a^0 from $\mathcal{A}$ based on ϵ,$\mathbf{Q}[s,a]_A$ and $\mathbf{Q}[s,a]_B$;
    observe reward $\mathcal{R}(s,a^0,s^0)$ and state s^0;
    select UPDATE ($\mathbf{Q}[s,a]_A$,$\mathbf{Q}[s,a]_B$);
    if *UPDATE*=$\mathbf{Q}[s,a]_A$ **then**
      update $Q(s,a)_1$ by (2) with A=1 and B=2;
    else
      update $Q(s,a)_1$ by (2) with A=2 and B=1;
    end
  end
  $s = s^0$
until $i = \max$ *RAO*;

Table I
THRESHOLDS FOR $\overline{N_{psu}}$

Traffic	*Config. 1*	*Config. 2*	*Config. 3*
Low	$\overline{N_{psu}} \leq 3$	$\overline{N_{psu}} \leq 5$	$\overline{N_{psu}} \leq 5$
Normal	$3 < \overline{N_{psu}} \leq 7$	$5 < \overline{N_{psu}} \leq 10$	$5 < \overline{N_{psu}} \leq 10$
High	$7 < \overline{N_{psu}} \leq 10$	$10 < \overline{N_{psu}} \leq 15$	$10 < \overline{N_{psu}} \leq 20$
Very High	$\overline{N_{psu}} > 10$	$\overline{N_{psu}} > 15$	$\overline{N_{psu}} > 20$

desired QoS. Given the trade-off between successful access probability and mean delay, and its relation to the amount of allowed contending UEs in the system, we have defined three different scenarios as follows. One where the priority is on accepting more UEs, another where the priority is to reduce the delay, and a last one that balances these two tendencies. Based on these principles, we have defined three different configurations for the reward function. Each configuration sets different thresholds for 4 traffic categories: Low, normal, high, and very high. These configurations are shown in Table I. The reward function sets a cost for states where traffic is low and P_{ACB} is low and vice versa. On the other hand, the reward function also penalizes states where $\Delta N_{psu} > 0$ or when $CV_{N_{psu}}$ is high. The full reward function is specified in [25].

V. PERFORMANCE EVALUATION

In this section, the Double Q-learning mechanism is evaluated testing its different configurations; we show how it can be adjusted according to the requirements of the network operator. The configuration for experimentation considers a single cell where H2H and M2M traffic coexists. By having this coexistence, it is possible to evaluate the impact that M2M UEs might have on H2H UEs. We used traces obtained from the Telco Telecom Italia [26] to represent the H2H traffic. This data was aggregated in periods of 10 minutes, and separated between SMS, voice, and data. However, the data only shows an intensity value without units, and therefore we pre-processed data, in such a way that the highest traffic represents 55 eRAB setups per second, which is a very high load on a single base station according to [27]. On the other hand, the traffic for M2M communications is represented by 30 000 UEs accessing the medium according to a Beta(3,4) distribution over 10 seconds as shown in [28]. This is a bursty traffic scenario, as it can occur in massive deployments of M2M communications such as those expected in IoT scenarios. This traffic model is commonly used in many access control studies. The main KPIs studied here are the mean access delay, the mean number of preamble transmissions, and the probability of successful access. These KPIs are commonly used in medium access studies, which allows the comparison with other solutions. The algorithm is trained with four days of data. In the case of H2H traffic, we use the traces whose values change every 10 minutes. In the case of M2M traffic we use the model explained earlier once every 10 minutes. For testing purposes, the H2H traffic is constant, with the highest possible load, that is, 55 calls/s. In all scenarios, the cellular network uses the PRACH configuration *prach-ConfigIndex 6*, [18], [28], with the parameter values listed in Table II. All the results shown are the mean values after 100 experiments. For the Double Q-learning algorithm, unless otherwise stated, the parameters are α =0.15 and γ =0.7. In the case of ϵ, its value starts in 0.9, and then it linearly decreases to zero.

Table II
RACH CONFIGURATION

Parameter	**Setting**
PRACH Configuration Index	*prach-ConfigIndex* $= 6$
Periodicity of RAOs	5 ms
Subframe length	1 ms
Available preambles for contention-based random access	$R = 54$
Maximum number of preamble transmissions	*preambleTransMax* $= 10$
RAR window size	$W_{\text{RAR}} = 5$ subframes
Maximum number of uplink grants per subframe	$N_{\text{RAR}} = 3$
Maximum number of uplink grants per RAR window	$N_{UL} = W_{\text{RAR}} \times N_{\text{RAR}} = 15$
Preamble detection probability for the kth preamble transmission	$P_d = 1 - \frac{1}{e^k}$ [28]
Backoff Indicator	$BI = 20$ ms
Re-transmission probability for *Msg3* and *Msg4*	0.1
Maximum number of *Msg3* and *Msg4* transmissions	5
Preamble processing delay	2 subframes
Uplink grant processing delay	5 subframes
Connection request processing delay	4 subframes
Round-trip time (RTT) of *Msg3*	8 subframes
RTT of *Msg4*	5 subframes

A. Evaluation of Double QL-ACB Mechanism

In this section, we compare the different configurations of our proposed Double Q-Learning solution with D-ACB [29], a well-known dynamic solution. In Fig. 3, the performance of the Double Q-Learning algorithm is shown after being trained with four days of data with the three different reward function configurations. Let us recall that the first configuration aims to avoid congestion by limiting the received number of

Table III
DOUBLE QL-ACB MECHANISM PERFORMANCE FOR DIFFERENT REWARD FUNCTIONS VS. D-ACB

KPI	*Config. 1*	*Config. 2*	*Config. 3*	*D-ACB*
$E(D)_{Total}$ (s)	7.9158	5.4129	4.2179	1.8801
$E(D)_{M2M}$ (s)	8.0837	5.5108	4.2986	1.8877
$E(D)_{H2H}$ (s)	3.8115	2.3533	1.7584	1.3852
$E(K)_{Total}$	1.71	1.95	1.97	4.49
$E(K)_{M2M}$	1.71	1.96	1.98	4.50
$E(K)_{H2H}$	1.58	1.68	1.66	3.65
$P_{sa-Total}$ (%)	99.99	99.89	99.36	78.91
P_{sa-M2M} (%)	99.99	99.89	99.35	78.77
P_{sa-H2H} (%)	100	99.97	99.80	89.60

preambles to 10, the second aims to limit the received number of preambles to 15, and the last one aims to limit the traffic at 20. From Table III, it can be seen that configuration 3 reduces the mean access delay considerably, although it has a lower successful access probability. In fact, as the traffic limit is set tighter, the successful access probability grows. In the same way, the mean number of preambles sent is also reduced. This is a symptom of lower congestion, and this is shown in Fig. 3. Clearly, configuration 1 has the lowest congestion (below 10), and configuration 3 has the highest. However, the peak in configuration 3 disappears earlier. This is of course an effect of the variation of P_{ACB}. As it is shown in Fig. 4, configuration 1 reduces P_{ACB} the most, that is, has a tighter constraint on traffic. Therefore, it is able to increase the successful access probability, while it increases the mean access delay. On the other hand, configuration 3 is not so strict, delaying less UEs, but reducing the successful access probability. However, in the worst case, this probability is reduced for all UEs to 99.36 %, that is, less than 1% of UEs will have to restart their connections. Although having to retry the connections to the network increases the total energy consumption of the devices, it is up to the network administrator to decide to which KPI to give more importance and adjust the reward function according to the service level agreements. This might be an attractive solution if we consider that in configuration 3 the mean access delay is reduced more than 3 s when compared to configuration 1 (7.91 s to 4.21 s). Also, it can be seen that the successful access probability of D-ACB is not as good as the Double Q-Learning solutions. In fact, more than 20 % of M2M UEs cannot access the system, far from what can be considered acceptable behavior. This is due to the fact that this solution causes many retries in the system, which is reflected in the higher values of K. Nevertheless, the UEs that can access the system suffer a considerably lower mean access delay.

In Fig. 5, the performance of the system with the three reward functions is illustrated when there are 40 000 M2M UEs. In this case, we have 30 % more traffic than the maximum established by the standards, and it can be seen that the performance for the reward functions 2 and 3 is poor, due to the high levels of collision, that can reach almost 300 in configuration 3. In fact, Table IV shows that the successful access probability for configurations 2 and 3 is below 80 %, although with some reductions on the mean access delay. Still, the performance of configuration 1 does not decay so sharply, since for an increase of 30 % of traffic, the probability of successful access is reduced in 3 %, and the delay is increased in 2 s. These values are not ideal, but show an important ability of our proposed mechanism to adapt to extraordinary traffic conditions. This is not what occurs with D-ACB, which shows a performance similar to that of configuration 3 in terms of successful access probability. Just like in the case of 30 000 M2M UEs, D-ACB increases the mean number of retransmissions which ultimately affects the overall performance of the system. However, D-ACB does not provide a mechanism that can be adapted like our Double Q-Learning solution.

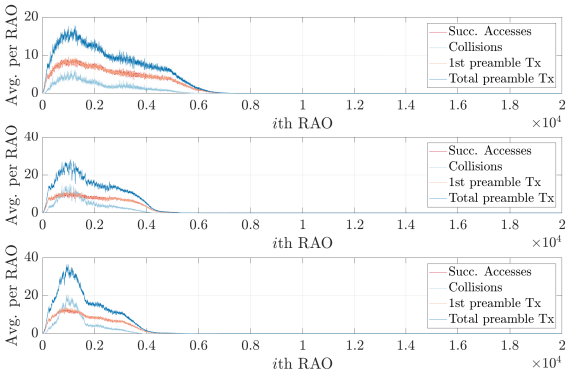

Figure 3. Congestion for configurations 1, 2, and 3.

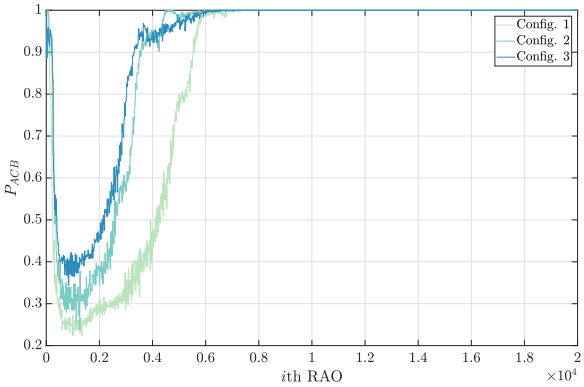

Figure 4. P_{ACB} variation for configurations 1, 2, and 3.

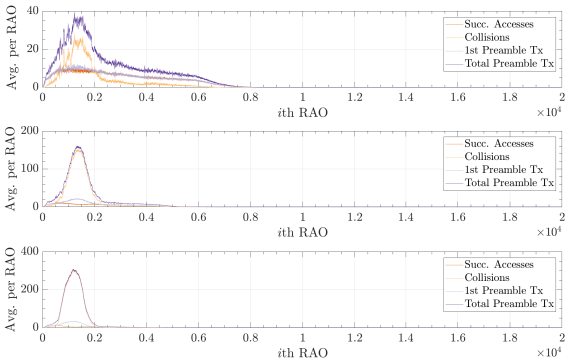

Figure 5. Congestion for configurations 1, 2, and 3 with 40 000 M2M UEs.

Table IV
DOUBLE QL-ACB MECHANISM PERFORMANCE FOR DIFFERENT REWARD FUNCTIONS AND 40 000 M2M UES VS. D-ACB

KPI	*Config. 1*	*Config. 2*	*Config. 3*	*D-ACB*
$E(D)_{Total}$ (s)	9.4617	5.0513	2.5578	2.1485
$E(D)_{M2M}$ (s)	9.6334	5.1418	2.6261	2.1598
$E(D)_{H2H}$ (s)	4.6155	2.1263	0.836	1.5711
$E(K)_{Total}$	1.86	2.29	2.66	4.94
$E(K)_{M2M}$	1.87	2.31	2.70	4.96
$E(K)_{H2H}$	1.62	1.72	1.70	4.06
$P_{sa-Total}$ (%)	96.79	72.73	45.74	45.88
P_{sa-M2M} (%)	96.74	72.42	45.07	45.54
P_{sa-H2H} (%)	98.74	89.00	80.56	75.01

VI. CONCLUSIONS

We have proposed a Double Q-Learning mechanism for LTE-A networks that dynamically adapts the barring rate of the ACB mechanism. The implementation of Double Q-learning allows reducing the risk of overestimating Q values, which is a common disadvantage of classical Q-Learning that might lead to sub-optimal performance. We have evaluated the mechanism in a scenario where both M2M and H2H communications coexist, and we have trained the system with real traces from H2H traffic. The results show that the mechanism can increase the probability of successful access and to diminish the mean number of preamble transmissions while insignificantly increasing the mean access delay. Since the ACB mechanism does not differentiate between H2H and M2M UEs, it impacts both types of UEs. To mitigate this impact, we have defined three different configurations for the reward function that allow the network operator to adjust the impact on the mean access delay or the successful access probability. We have evaluated each configuration, and the results show that an administrator can weight on the trade-off of the different KPIs by choosing the appropriate reward function configuration, which is an essential characteristic for real deployments of the solution. Also, we compared our solution against D-ACB, a well-known dynamic solution, and the results reveal that our proposed solutions perform better under different traffic conditions.

ACKNOWLEDGMENT

The research of D. Pacheco-Paramo was supported by Universidad Sergio Arboleda, under project "Plataforma de datos para territorios inteligentes," IN.BG.086.19.014. The research of L. Tello-Oquendo was conducted under project CONV.2018-ING010, Universidad Nacional de Chimborazo.

REFERENCES

[1] Ericsson. (2017, Nov.) Ericsson mobility report. [Online]. Available: https://www.ericsson.com/mobility-report

[2] Cisco. (2017, Mar.) Cisco visual networking index (VNI): Global mobile data traffic forecast update, 2016-2021. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[3] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, February 2017.

[4] 3GPP, *TS 22.368, Service Requirements for Machine-Type Communications*, Dec 2014.

[5] ——, *TS 22.011, V15.1.0, Service Accessibility*, June 2017.

[6] L. Tello-Oquendo, I. Leyva-Mayorga, V. Pla, J. Martinez-Bauset, J.-R. Vidal, V. Casares-Giner, and L. Guijarro, "Performance analysis and optimal access class barring parameter configuration in LTE-A networks with massive M2M traffic," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3505–3520, 2018.

[7] A. Lo, Y.-W. Law, M. Jacobsson, and M. Kucharzak, "Enhanced LTE-advanced random-access mechanism for massive machine-to-machine (M2M) communications," 2011.

[8] R.-H. Hwang, C.-F. Huang, H.-W. Lin, and J.-J. Wu, "Uplink access control for machine-type communications in lte-a networks," *Personal and Ubiquitous Computing*, vol. 20, no. 6, pp. 851–862, Nov 2016.

[9] L. Tello-Oquendo, J.-R. Vidal, V. Pla, and L. Guijarro, "Dynamic access class barring parameter tuning in lte-a networks with massive m2m traffic," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. IEEE, 2018, pp. 1–8.

[10] H. Kim, S. s. Lee, and S. Lee, "Dynamic extended access barring for improved M2M communication in LTE-A networks," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2017, pp. 2742–2747.

[11] C. M. Chou, C. Y. Huang, and C.-Y. Chiu, "Loading prediction and barring controls for machine type communication," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, jun 2013, pp. 5168–5172.

[12] L. Tello-Oquendo, D. Pacheco-Paramo, V. Pla, and J. Martinez-Bauset, "Reinforcement Learning-Based ACB in LTE-A Networks for Handling Massive M2M and H2H Communications," in *Proc. IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[13] J. Moon and Y. Lim, "A Reinforcement Learning Approach to Access Management in Wireless Cellular Networks," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–17, 2017.

[14] A. S. El-Hameed and K. M. Elsayed, "A Q-learning approach for machine-type communication random access in LTE-Advanced," *Telecommunication Systems*, pp. 1–17, 2018.

[15] L. M. B. P. M. D. Grace, "Q-learning Based Random Access with Collision free RACH Interactions for Cellular M2M," in *Proc. 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, Sep 2015.

[16] 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.

[17] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[18] 3GPP, *TS 36.321, Medium Access Control (MAC) Protocol Specification*, Sept 2012.

[19] ——, *TS 36.211, Physical Channels and Modulation*, Dec 2014.

[20] L. Tello-Oquendo, V. Pla, I. Leyva-Mayorga, J. Martinez-Bauset, V. Casares-Giner, and L. Guijarro, "Efficient Random Access Channel Evaluation and Load Estimation in LTE-A with Massive MTC," *IEEE Transactions on Vehicular Technology*, pp. 1–5, 2018.

[21] 3GPP, *TS 36.331, Radio Resource Control (RRC), Protocol specification*, Sep 2017.

[22] ——, *TR 36.912, Feasibility study for Further Advancements for E-UTRA*, Apr 2011.

[23] H. V. Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 2613–2621.

[24] C. J. C. H. Watkins and P. Dayan, "Technical Note Q-Learning," *Machine Learning*, vol. 8, pp. 279–292, 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[25] D. Pacheco-Paramo and L. Tello-Oquendo. (2019, Jul.) Adjustable Reward Function for Double Q-Learning ACB Solution. [Online]. Available: https://dfpp.co/documents

[26] Telecomitalia. (2016, Nov.) Telecom italia: Big data challenge. [Online]. Available: http://www.telecomitalia.com/tit/en/innovazione/archivio/big-data-challenge-2015.html

[27] Nokia, "Mobile Broadband solutions for Mass Events," Nokia, Tech. Rep., 2014.

[28] 3GPP, *TR 37.868, Study on RAN Improvements for Machine Type Communications*, Sept 2011.

[29] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9847–9861, 2016.