Reinforcement Learning Based Preamble Resource Allocation Scheme for Access Control in Machine-to-Machine Communication

Hongyu Liu[1](✉), Bei Liu[1,2], Hui Gao[3], Xibin Xu[4], and Xin Su[4]

[1] School of Communication and Information Engineering,
Chongqing University of Posts and Telecommunications,
Chongqing, China
`S200131182@stu.cqupt.edu.cn`
[2] Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing, China
`liubei@mail.tsinghua.edu.cn`
[3] Beijing University of Posts and Telecommunications, Beijing, China
`huigao@bupt.edu.cn`
[4] Department of Electronic Engineering Tsinghua University, Beijing, China
{`xuxb,suxin`}`@tsinghua.edu.cn`

Abstract. With the rapid development of Internet of Things (IoT) technology, the number of large numbers of Machine Type Communication (MTC) devices involved in M2M has increased dramatically. When large scale MTC devices access the base station at the same time in a short period of time, this can cause traffic overload and lead to a sharp drop in the success rate of access of MTC devices. 3GPP has proposed the access class barring (ACB) scheme to defer access requests from certain activated MTC devices to avoid congestion at the base station (BS). In this paper, we propose a dynamic ACB scheme for grouping MTC devices and a resource allocation scheme for preamble. First, MTC devices are classified into two categories according to their characteristics: delay-sensitive and energy-constrained. The two categories use separate preamble resources, and a temporary ACB factor is calculated for each time slot based on the current preamble resources and the number of devices. The preamble resources are reallocated based on this temporary ACB factor using reinforcement learning methods, and then the ACB factor is dynamically adjusted according to the new preamble resources. Simulation results show that the solution improves the access success rate of M2M devices, reducing the total service time of delay-sensitive devices by 40% compared to the traditional solution, while reducing the access collision rate of energy-constrained devices by 30%.

Keywords: Machine-to-machine communications · Random Access Control · Q-learning

---

Supported by the National Key R&D Program of China under Grant 2020YFB1806702.

F. Gao et al. (Eds.): ChinaCom 2022, LNICST 500, pp. 44–56, 2023.
https://doi.org/10.1007/978-3-031-34790-0_4

1 Introduction

The widespread adoption of the IoT [1] in various fields such as industry, agriculture, healthcare and transportation has stimulated an explosive growth in the number of IoT devices. With the growth of the IoT devices, M2M communication is expected to become one of the main drivers of cellular networks [2,3] M2M communication, also known as MTC, refers to communication between machine devices without human intervention. Many devices can be triggered almost simultaneously and attempt to access a base station via a random access channel (RACH), which can cause a surge in M2M traffic. M2M communication, as the key part of IoT development, is essential for efficient data transfer from machines and devices to the network for various IoT applications such as smart metering, healthcare, smart appliances, surveillance, security and logistics tracking. Studies from 3GPP and the literature have shown that the physical random access channel (PRACH) of a cell can be severely overloaded when tens of thousands of MTDs wake up and attempt to access the cell in a highly synchronised manner. However, traditional access control methods may not be able to handle overload in the radio access network (RAN) and core network (CN), which can lead to severe congestion and delays. The problem becomes worse if a faulty device keeps trying to access the network, resulting in a large proportion of machine communication devices not being able to gain access before the maximum number of allowed attempts is exceeded. The ACB scheme is easy to implement and is very effective for congestion control, therefore it is defined as a viable scheme by the Radio Resource Control specification. In the first step of the RACH process, the base station broadcasts an ACB factor p to all UEs before the start of each time slot via system messages, with p taking values from 0 to 1. In the same time slot, each UE generates a random number between 0 and 1 and compares it with the value of p before attempting to access the PRACH channel. If the random number is less than the broadcast ACB factor p, the UE device will continue to access the base station. Otherwise, the UE device is denied access at this time slot and the process is repeated at the next time slot or after being blocked for a certain amount of time. In this way, the ACB reduces the number of access requests per time slot.

The size of the ACB factor is fixed in traditional ACB schemes, which can significantly improve access success rates in large-scale burst access scenarios, but at the cost of a dramatic increase in access latency. We consider a large-scale IoT scenario such as the smart city in Fig. 1, where devices are divided into delay-sensitive devices and energy-constrained devices. For example, self-driving cars need to communicate with nearby vehicles with low latency, while fixed devices in the surrounding environment, such as surveillance cameras, have low data rates and infrequent transmissions. The size of the signalling packets used in the wireless network to synchronise sensor devices to the base station or to resolve contention between sensor devices can be much larger than the size of the user data packets used for sensor transmissions. Therefore, for the sensor class of devices, where energy constraints are a key feature, access collisions should be avoided as much as possible. When these two types of devices coexist, the traditional ACB scheme with a fixed ACB factor is clearly not suitable for this scenario, and the traditional ACB scheme does not take good care of the access characteristics of both types of devices.

Fig. 1. A massive IoT network

To solve the above problems, this paper proposes an access control strategy and a dynamic preamble grouping scheme based on reinforcement learning that is applicable to the coexistence scenario of two types of devices. The main contributions of this paper are summarized as follows.

1. We divide MTCD into two groups according to realistic scenarios, while dynamically assigning the preamble resources to the two types of devices, which complete the random access process through the ACB mechanism.
2. We propose a method for preamble resource allocation through reinforcement learning. A temporary ACB factor is calculated in each time slot based on the current resources, and the preamble resources are reallocated based on this temporary ACB factor to enable both types of devices to use the preamble resources efficiently.
3. Based on the derived values of ACB factors for delay-sensitive devices in related work, the ACB factors for energy-constrained devices are derived and the optimal ACB factors are calculated based on the number of activated devices and the size of the leading resources.
4. Simulation results show that the scheme results in lower collision rates for energy-constrained devices, reduced congestion conflicts, and significantly lower total service times for delay-sensitive devices.

The rest of this paper is organized as follows Sect. 2 provides an initial introduction to the work related to random access control and the random access process. In Sect. 3, we detail the design of our proposed scheme. In Sect. 4, the performance of our proposed scheme is evaluated by comparing it with a typical conventional scheme. Section 5 contains concluding remarks.

2 Related Work

To improve the latency performance of traditional ACB schemes while maintaining their access success rate, various works have proposed dynamic adjustment of the ACB factor over time. The article [4] uses the QOE of device delay mapping as the objective function for reinforcement learning. Due to the different device characteristics, different delay mapping functions can better differentiate devices and achieve priority access to low latency devices. The article [5] proposes a dual-Q learning-based ACB mechanism to determine the ACB parameters, which is able to dynamically adapt to different traffic conditions when M2M and H2H communications coexist, with three different configurations allowing a trade-off between the probability of successful access and the average access latency. The article [6] proposes a dynamic ACB approach where the value of the ACB factor changes adaptively within each time slot depending on the service load. In order to find the optimal ACB factor the article proposes a method to estimate the number of devices based on the information obtained by the base station. The article [7] proposes the use of a reinforcement learning method called dueling deep Q networks to dynamically adjust the ban factor and average ban time. Computer simulations show that for a given access delay and energy consumption tolerance, our design can achieve significantly higher energy satisfaction while maintaining comparable delay satisfaction compared to schemes that focus only on restriction factor adjustment. The article [8] presents a new deep reinforcement learning algorithm that is first used to dynamically adjust the ACB factor in a uniform priority network. The algorithm is then further enhanced to adapt to different MTCDs with different quality of service (Qos) requirements. article [9] combines ACB and extended access limits (EAB) to increase access latency while reducing the conflict probability, resulting in higher energy efficiency. With respect to the allocation of leading resources, the article [10] determines the optimal value of the ACB factor to reduce traffic overload in the ideal case where the eNodeB knows the number of backlogged MTC devices. To make better use of the random access resources shared between human users and MTC devices in LTE networks, methods are proposed to dynamically allocate the preamble resources for MTC devices. Most dynamic ACB solutions focus only on tuning the ACB factor, and the derivation of the ACB factor is based on reducing the latency as well as the total service time of the system, with no consideration given to energy-constrained devices.

3 System Model

3.1 Random Access Process

In current cellular networks, UEs need to perform random access processes to achieve uplink synchronisation with the base station and to obtain radio channel access resources. The two types of random access are competition-based and non-competition-based. This paper focuses on the contention-based random access process. This process consists of four steps.

1. Preamble (MSG1): In the first step, all UEs that pass the ACB test randomly select a preamble and transmit it on the physical random access channel.
2. RAR (MSG2): In the second step, the base station confirms that all the preamble have been successfully received using the Random Access Request Response (RAR), which contains the identification of the detected preamble and the uplink permission for the Step 3 message MSG3.

3. Connection request (MSG3): In step 3, the UE sends the MSG3 of its ID in the PUSCH after receiving its corresponding RAR within the random access response window size time. When two UEs have selected the same preamble in step 1, both UEs will be granted the same block of time-frequency resources for uplink transmission of MSG3, when a conflict occurs.
4. Contention resolution (MSG4): In step 4, the base station broadcasts the contention solution containing the IDs of the UEs that have successfully decoded MSG3 without any response to the conflicting MSG3s and declares access failure in the contention solution for those UEs that have selected the conflicting preamble.

3.2 System Traffic Model

We assume that 54 preamble are available and that there are a total of N MTC devices in a cell. As these devices are not synchronised, they will not be active at the same time, but will be active for a short T_A period. We refer to this time as the activation time. Each device is activated with probability $f(t)$ over a time horizon of length T_A. The popular choice for modelling service volume bursts proposed in 3GPP is the beta distribution with the following expression for $f(t)$,$B(\alpha, \beta)$ as a function of β, where $\alpha = 3, \beta = 4$.

$$f(t) = \frac{t^{\alpha-1}(T_A - t)^{\beta-1}}{T_A^{\alpha+\beta-1} B(\alpha, \beta)}, 0 \leq t \leq T_A \tag{1}$$

In this system model, time is divided into consecutive time slots, each represented by an integer $i = 1, 2 \cdots\cdots$. Thus T_A can be divided into I_A time slots, with I_A denoting the number of $RACH$ in the T_A time range, where the duration of the ith time slot is from the t_{i-1} moment to t_i. It is assumed that the activation of the MTC device is completed at the beginning of the time slot. Denote the number of newly activated devices in time slot i by λ_i, where $i = 1, 2 \cdots\cdots I_A$. The λ_i depends specifically on the distribution of the activated traffic $f(t)$ and the total number of devices N, with the expression

$$\lambda_i = N \int_{t_{i-1}}^{t_i} f(t) dt, i = 1, 2 \cdots\cdots I_A \tag{2}$$

4 Proposed Preamble Allocation and Access Control Scheme

The number of MTC devices that need to be accessed in each time slot plays a crucial role in the design of the ACB scheme. The number of MTC devices that need to be accessed in each time slot is unknown to both base stations and terminals because of the inconsistent activation times of MTC devices. A number of methods have been proposed by a large number of research institutions to estimate the number of devices in a time slot, and the estimated number of devices in [4] is very close to the actual number of devices. In this paper, the number of MTC devices to be accessed in each time slot is known by default and the ACB scheme for two types of devices is proposed.

4.1 Dynamic Adjustment of ACB Factor

In the scenario we set up, assuming $M = 54$ preamble are available and a total of N MTC devices are active at time T_A, we classify these N devices into two categories: delay-sensitive (DSD) and energy-constrained (ECD). Our ACB scheme is different for different types of devices, and the number of these two types of devices is N_D and N_E. So we have

$$N_D + N_E = N \tag{3}$$

Correspondingly, the number of preamble owned by the two types of devices are M_D and M_E respectively.

$$M_D + M_E = M \tag{4}$$

ACB Factor for Delay-Sensitive Devices. Delay-sensitive devices need to be connected to the network as soon as possible, so the total service time needs to be reduced. The total service time is defined as the number of time slots consumed between the activation of the device and the successful transmission of the preamble. Given the number of access devices $N_i = n$ at each time slot, the ideal ACB factor for the minimum total service time according to article [10] is

$$p_D = min(1, \frac{M_D}{n_D}) \tag{5}$$

ACB Factor for Energy-Constrained Devices. What needs to be considered for this class of devices is the energy consumed during the whole random access process because of the energy constraint. When multiple devices select the same preamble, the base station detects a preamble collision, and the device that selected the preamble fails this access and needs to initiate random access again after waiting for an avoidance interval. When the probability of preamble collision is high, the device needs to initiate random access several times, which will consume a lot of energy because of the uplink transmission. Therefore, for energy-constrained devices, our solution focuses on reducing their collision probability.

In time slot i, it is assumed that n MTC devices randomly select one of the preamble sequences with equal probability. The total number of preamble is M and the probability of selecting preamble m is $1/M$. Let $P_m = 0, P_m = 1, P_m > 1$ denote the three cases of unselected preamble, successful transmission and collision occurring instead.

The probability that the preamble m is not selected is

$$P(P_m = 0) = (1 - \frac{1}{M})^n \tag{6}$$

The probability of successful transmission of the preamble m is

$$P(P_m = 1) = C_n^1 \times \frac{1}{M} \times (1 - \frac{1}{M})^{n-1} \tag{7}$$

The probability of a collision of the preamble m is

$$P(P_m > 1) = 1 - P(P_m = 0) - P(P_m = 1) = 1 - (1 - \frac{1}{M})^n - \frac{n}{M} \times (1 - \frac{1}{M})^{n-1} \quad (8)$$

Let C_i be the number of conflicting preamble in time slot i and p be the ACB factor for that time slot. Since the collision and non-collision events for each preamble are independent and identically distributed binomial distributions, the expectation of the number of conflicting preamble can be found as

$$\begin{aligned} E(C_i) = \sum_{m=1}^{M} P(P_m > 1) = M \times \left[1 - (1 - \frac{1}{M})^{np} - \frac{np}{M} \times (1 - \frac{1}{M})^{np-1}\right] \\ \approx M - (M + np)e^{-\frac{np}{M}} \end{aligned} \quad (9)$$

Similarly let S_i be the number of preamble successfully transmitted in time slot i as

$$E(S_i) = \sum_{m=1}^{M} P(P_m = 1) = np \times (1 - \frac{1}{M})^{np-1} \approx np \times e^{-\frac{np}{M}} \quad (10)$$

The ACB factor for delay-sensitive devices is derived by maximising the access success preamble per time slot, and the number of collisions needs to be considered for energy-constrained devices, so calculating the ACB factor for energy-efficient devices requires a combination of the above two equations to consider.

$$E(X_i) = E(S_i) - E(C_i) = (M + 2np) \cdot e^{-\frac{np}{M}} - M \quad (11)$$

Derive for p

$$\frac{d}{dp} E(X_i) = (n - \frac{2n^2 p}{M}) \cdot e^{-\frac{np}{M}} \quad (12)$$

Letting the equation be 0, we obtain $p = \frac{M}{2n}$. So the ACB factor for the energy-constrained device is set to $\frac{M}{2n}$.

4.2 Reinforcement Learning Based Preamble Resource Allocation

The number of preamble per time slot is finite, and the ultimate goal of both device types is to successfully access the network; both access strategies cater for more devices to access while ensuring their access characteristics. We consider the use of reinforcement learning to solve the problem of preamble allocation, as it enables the use of computer simulations to generate reasonable solutions without the need to build complex theoretical models that allow devices to use the least amount of leading resources while maintaining the minimum service time. the ACB factor maximises the use of the limited number of preamble for access control to minimise the total service time for the delay-sensitive class of devices and the energy-constrained devices access characteristics. Therefore, the allocation of preamble is an efficient use of resources, reducing the number of preamble when the number of device activations is low and increasing the number of preamble when the number of device activations is high, focusing on the access of latency-sensitive devices in the process.

Let S denote a finite set of possible environment states and let A denote a finite set of admissible actions to be taken. At RA slot t, BS perceives the current state $s_t = s \in S$ of the environment and take an action $a_t = a \in A$ based on both the perceived state and its past experience. The action a_t changes the environment state from s_t to $s_{t+1} = s^{'} \in S$. When that happens, the system receives the reward r_t. The goal of the system is to find an optimal strategy The goal of the system is to find an optimal strategy.

$$\pi^*(s) = \underset{a \in A}{\operatorname{argmax}}(Q^*(s,a)) \tag{13}$$

$Q^*(s,a)$ is the optimal Q value, which is defined as

$$Q_t(s,a) = Q_t(s,a) + \alpha(r_t(s,a) + \gamma_t \max_{a_{t+1} \in A} Q_t(s_{t+1}, a_{t+1}) - Q_t(s,a)) \tag{14}$$

$\alpha(0 \leq \alpha \leq 1)$ is the learning rate and $\gamma(0 \leq \gamma \leq 1)$ is the discount rate. When the learning rate is 0, the Q value is never updated. When the learning rate is set to higher, learning will be rapid and the discount factor will be weighted more heavily on the current reward than the reward.

In this paper, we build a QL algorithm to allocate preamble resource to increase the number of successful accesses to delay sensitive devices in each time slot. The state space consists of the ACB factor of the current time slot. The size of the ACB factor is related to the number of preamble and the number of devices to be accessed. The ACB factor calculated from the state of the current pool of preamble and the number of devices in the current time slot is used as the state, and the action is selected to update the pool of preamble, and the recalculated ACB factor is the ACB factor of the current time slot. The size of the ACB factor as a state is discrete into intervals $(0, 0.01), (0.01, 0.1), (0.1, 0.3), (0.3, 0.5), (0.5, 0.7), (0.7, 1)$, when P is large, indicating that the number of devices is small or the number of preamble is assigned high at this time, so that the number of preamble can be reduced to another class of devices. The action space is represented by A. The number of preamble for delay-sensitive device is incremented or decremented to adjust the allocation of preamble resources for each time slot, and these actions are decremented or incremented by $\delta_i(\delta_i \in \{-10, -7, -5, -3, -1, 0, +1, +3, 5, +7, +10\})$ or kept at their current values. To balance the use of learning and exploration, the QL algorithm uses the $\epsilon - greey$ method. When an action needs to be selected, the BS primarily selects the action a with the largest $Q(s,a)$ in state s. The BS randomly selects the action a with probability ϵ from the allowed actions in state s.

The reward function focuses on the number of successful accesses to delay-sensitive devices, reducing the restriction on delay-sensitive devices by assigning them more leading resources in the presence of network congestion, and the relative energy-limited because of the reduction in leading resources, the ACB will be more restrictive. The number of successful delay-sensitive devices and the number of successful energy-constrained devices are weighted and summed, and then divided by the total number of preamble M-normalised

$$r_t(s,a) = \frac{\omega N_L^t + (1-\omega) N_E^t}{M} \tag{15}$$

N_L^t is the number of delay-sensitive devices successfully accessed, N_E^t is the number of energy-constrained devices successfully accessed, and ω is the smoothing factor.

Algorithm 1: RL-based preamble resource allocation algorithm

```
Input: The number of current time slot devices and the number of
       preamble are used to obtain a temporary ACB factor according to
       (5)(12)
Output: The number of increments and decrements in the preamble of
        delay-sensitive devices
Initialise the number of preamble available to the device;
for episode = 0, 1, 2, ... do
    while Number of devices connected $n <$ Total number of devices $N$ do
        Calculation of a temporary ACB factor based on the number of
        devices that currently need to be accessed and the number of
        preamble;
        Select an action $a^n = i$, $i \in$ A based on the greedy policy, where i
        represents the number of increments and decrements in the
        preamble;
        Calculate the ACB factor for this time slot according to (5)(12);
        $n+ =$ Number of successfully accessed devices;
    end
end
```

5 Performance Evaluation

In this section, we validate the effectiveness of the proposed scheme in terms of both the collision rate of energy-constrained devices and the total service time of both types of devices.

5.1 Comparison of Collision Rates for ECD

The collision rate refers to the ratio of the number of devices with failed access at time slot t to the number of devices initiating access. Figure 2 shows the ACB

Table 1. RACH Configuration

Parameter	Setting
Preamble number M	54
Total number of M2M devices N	30000
DSD number	10000
ECD number	20000
Backoff indicator	20 ms

factor for DSDs is set with the goal of the shortest total service time and lowest latency, and does not take into account the collision rate. When an ECD uses a dedicated ACB factor, the collision rate is reduced compared to sharing the same ACB factor with a low-latency device. The reduced collision rate means that the ECD initiates accesses less often, saving the energy required to initiate accesses.

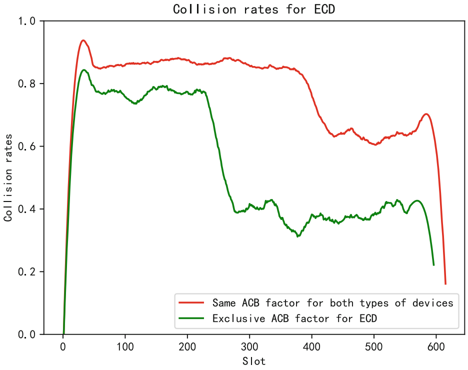

Fig. 2. Collision rate per time slot for ECD

5.2 The Impact of Preamble Resource Allocation on the Probability of Access Success

When DSD and ECD coexist, the percentage of preamble allocated affects access performance. In scenarios where the proportion of DSDs is 10%, 20% and 30% respectively, Fig. 3 shows that when the proportion of DSDs is 10%, the number of successfully accessed devices first increases and then decreases as the number of preamble allocated to DSDs increases. The analysis leads to the conclusion that the optimal allocation of the preamble is different for different proportions of DSDs. In a practical scenario, after estimating the actual number of DSDs and ECDs, the optimal allocation of preamble can be chosen to obtain the optimal probability of successful access.

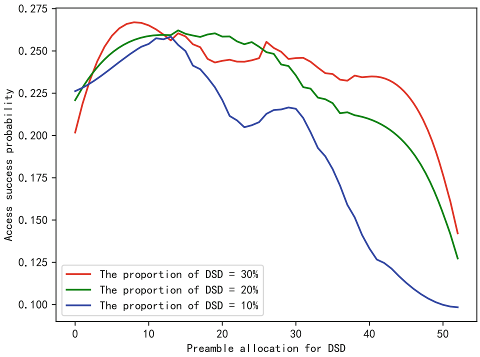

Fig. 3. Different preamble allocation affects the access success probability

5.3 The Impact of the Number of Preamble Resources on the Total Service Time

The objective of the solution proposed in this paper is to reduce the total service time of DSD on a basis that does not affect the total service time of all devices. The schemes shown in Fig. 4 are two types of devices sharing a pool of preamble, a fixed ratio of preamble, a dynamically adjusted ratio of preamble as proposed in the paper [11], and the reinforcement learning-based preamble resource allocation scheme proposed in this paper. It can be seen that the proposed scheme DSD in this paper has the lowest total service time when the total service time of all devices is similar.

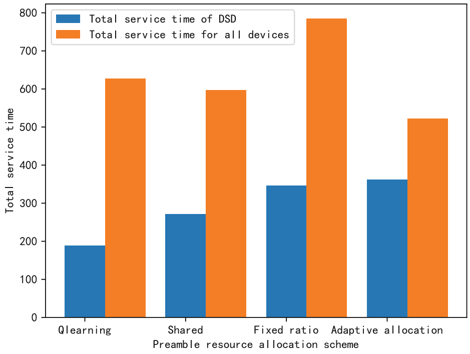

Fig. 4. The impact of the preamble resource allocation scheme on total service time

6 Conclusion

In this paper, we focus on the coexistence of large-scale DSD and ECD in cellular communication systems, and propose an access control strategy and a dynamic preamble grouping scheme based on reinforcement learning for the coexistence of the two types of devices, taking full account of the QoS requirements of DSDs, and dividing the preamble into two groups. In this paper, we use the current ACB factor for each time slot to allocate the preamble resources by reinforcement learning, and the ACB factor is dynamically updated according to the number of preamble and the number of devices. The paper verifies the feasibility and effectiveness of the scheme using Python simulations. The collision rate of energy-limited devices is reduced by 30%, while dynamically adjusting the number of preamble has a significant effect on reducing the total service time of DSD devices by 40% compared to the conventional scheme.

References

1. Nguyen, D.C., Ding, M., Pathirana, P.N., Seneviratne, A., Li, J., Niyato, D., Dobre, O., Poor, H.V.: 6g internet of things: A comprehensive survey. IEEE Internet Things J. **9**(1), 359–383 (2022). https://doi.org/10.1109/JIOT.2021.3103320
2. Wu, H., Zhu, C., La, R.J., Liu, X., Zhang, Y.: Fasa: Accelerated s-aloha using access history for event-driven m2m communications. IEEE/ACM Trans. Netw. **21**(6), 1904–1917 (2013). https://doi.org/10.1109/TNET.2013.2241076
3. Laya, A., Alonso, L., Alonso-Zarate, J.: Is the random access channel of lte and lte-a suitable for m2m communications? a survey of alternatives. IEEE Commun. Surv. Tutor. **16**(1), 4–16 (2014). https://doi.org/10.1109/SURV.2013.111313.00244
4. Zhang, D., Liu, J., Zhou, W.: Acb scheme based on reinforcement learning in m2m communication. In: GLOBECOM 2020–2020 IEEE Global Communications Conference, pp. 1–6 (2020). https://doi.org/10.1109/GLOBECOM42002.2020.9322144
5. Pacheco-Paramo, D., Tello-Oquendo, L.: Adjustable access control mechanism in cellular mtc networks: A double q-learning approach. In: 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), pp. 1–6 (2019). https://doi.org/10.1109/ETCM48019.2019.9014871
6. Haider Shah, S.W., Riaz, A.T., Iqbal, K.: Congestion control through dynamic access class barring for bursty mtc traffic in future cellular networks. In: 2018 International Conference on Frontiers of Information Technology (FIT), pp. 176–181 (2018). https://doi.org/10.1109/FIT.2018.00038
7. Bui, A.T.H., Pham, A.T.: Deep reinforcement learning-based access class barring for energy-efficient mmtc random access in lte networks. IEEE Access **8**, 227657–227666 (2020). https://doi.org/10.1109/ACCESS.2020.3045811
8. Chen, Z., Smith, D.B.: Heterogeneous machine-type communications in cellular networks: Random access optimization by deep reinforcement learning. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–6 (2018). https://doi.org/10.1109/ICC.2018.8422775
9. Toor, W.T., Jin, H.: Combined access barring for energy and delay constrained machine type communications. In: 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp. 130–132 (2018). https://doi.org/10.1109/ICTC.2018.8539640

10. Duan, S., Shah-Mansouri, V., Wang, Z., Wong, V.W.S.: D-acb: Adaptive congestion control algorithm for bursty m2m traffic in lte networks. IEEE Trans. Veh. Technol. **65**(12), 9847–9861 (2016). https://doi.org/10.1109/TVT.2016.2527601
11. Zhao, X., Wang, C., Wang, W.: Dynamic preamble grouping and access control scheme in machine-to-machine communication. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6 (2019). https://doi.org/10.1109/WCSP.2019.8928049