Delay-aware Priority Access Classification for Massive Machine-type Communication

Mayukh Roy Chowdhury and Swades De

***Abstract*—Massive Machine-type Communications (mMTC) is one of the principal features of the 5th Generation and beyond (5G+) mobile network services. Due to sparse but synchronous MTC nature, a large number of devices tend to access a base station simultaneously for transmitting data, leading to congestion. To accommodate a large number of simultaneous arrivals in mMTC, efficient congestion control techniques like access class barring (ACB) are incorporated in LTE-A random access. ACB introduces access delay which may not be acceptable in delay-constrained scenarios, such as, eHealth, self-driven vehicles, and smart grid applications. In such scenarios, MTC devices may be forced to drop packets that exceed their delay budget, leading to a decreased system throughput. To this end, in this paper a novel delay-aware priority access classification (DPAC) based ACB is proposed, where the MTC devices having packets with lesser leftover delay budget are given higher priority in ACB. A reinforcement learning (RL) aided framework, called DPAC-RL, is also proposed for online learning of DPAC model parameters. Simulation studies show that the proposed scheme increases successful preamble transmissions by up to 75% while ensuring that the access delay is well within the delay budget.**

***Index Terms*—Massive machine-type communication, delay-sensitive, priority access classification, random access, access class barring, reinforcement learning**

I. Introduction

MACHINE to Machine (M2M) communication or Machine-type Communication (MTC) refers to the technology or framework where intelligent machines communicate among themselves with little or no human intervention [1]. Massive Internet of Things (IoT) or Massive Machine-Type Communication (mMTC) refers to a large number of such autonomous machines connected in a network. MTC applications include but not limited to smart metering, payment, object tracking, remote surveillance, e-health [2], [3]. mMTC is the core technology in modern infrastructures, such as smart cities, smart grid, and industry 4.0 [4].

MTC is in many ways different from conventional Human to Human (H2H) communication, for which cellular network standards like Long Term Evolution (LTE) and LTE - Advanced (LTE-A) were designed. In LTE/LTE-A, random access is mainly used to achieve two targets: uplink synchronization and getting radio resources for sending higher-layer messages. Upon powering up, an MTC device initiates the registration mechanism by sending Radio Resource Control (RRC) connection request to the nearest base station (evolved Node B or eNB). Random access in LTE can be categorized into two: (i) contention-based and (ii) contention-free [5]. In this paper the contention-based variant is considered.

In contention-based approach, MTC devices contend for getting access to the eNB. Unique quality of service (QoS) parameters of the MTC devices pose various challenges in the existing LTE/LTE-A infrastructure [6]. One such challenge arises due to a unique traffic pattern of mMTC and the limited number of preambles available with the eNB. Usually the MTC devices communicate among each other or to remote cloud servers for data transmission through the eNBs at certain periodicity. When a huge number of devices attempt to access an eNB at the same time, random access preamble collision increases. Hence, effective random access algorithms are required to control the access to eNB so that maximum possible devices are given access. Access Class Barring (ACB) was suggested by 3GPP for tackling Radio Acess Network overload in LTE as well as in Narrowband IoT (NB-IoT) [2], [7]. ACB redistributes the devices over a longer period of time; in the process it introduces considerable delay. Therefore, to accurately evaluate efficiency of random access, it is very important to consider the delay encountered by the MTC devices, especially in the delay-constrained scenarios [8].

A. Related works

Complex architecture and diverse design parameters of random access procedure makes it hard to evaluate its performance analytically. Hence, in most of the existing literature, its performance has been evaluated through simulations and iterative methods. Yet, there is a consensus that the current LTE random access is not suitable to deal with massive MTC scenario [6], [9]–[11]. Among the few works that have analytically modeled the random access procedure [12]–[14], most of them lacked in accuracy owing to different reasons. The authors in [12] modeled the random access delay, while in [15] a lower bound of access delay was obtained by triangular approximation of the beta distribution used for modeling arrivals. However, these works did not consider any access control technique like ACB. In [13], both random access and waiting delays due to extended access barring (EAB) were modeled with the access delay model from [12]. These models suffer from low accuracy. On the other hand, some of the works [14], [16] considered only average random access delay, which may not always be useful. What can be of more significance is delay characterization due to ACB. The authors

This work was supported by the Science and Engineering Research Board, DST, under Grant CRG/2019/002293.

M. Roy Chowdhury and S. De are with the Department of Electrical Engineering and Bharti School of Telecom., IIT Delhi, New Delhi, India.

in [17] presented an analytical model of ACB, which accounts for the details of LTE-A specification. They also characterized the access delay due to ACB and derived the delay probability mass function (PMF) using an iterative algorithm.

Different ACB schemes have been proposed in recent literature [18]–[20]. Some standard configuration with a range of allowed values of relevant parameters are given in 3GPP LTE/LTE-A standards document [9]. But these specifications do not state the optimum barring rates for the different access classes that would maximize the system utility. A dynamic ACB algorithm, called D-ACB, in [19] proposed to dynamically tune the barring rate of MTC devices in each random access opportunity (RAO) to maximize preamble transmission throughput. However, D-ACB did not address delay sensitivity of the packets and assigned same barring rate to all MTC devices. An analysis of ACB performance was presented in [10], where the distinctive feature is that, instead of following only typical barring rate values used in the prior works, they can take any value within the range suggested by 3GPP. Also, unlike most of the prior works, the approach in [10] does not consider barring duration to be constant. A prioritized access scheme, named PRADA, was proposed in [21], where random access channel resources are pre-allocated to different classes of devices. EAB proposed in 3GPP LTE-A controls MTC traffic congestion by disallowing access to the low priority MTC devices based on a barring bitmap [13]. Performance of two of the most popular access control schemes, ACB and EAB, were compared in [22], and it was noted that ACB works better in delay-sensitive MTC applications.

Learning based approaches have been used to tackle access congestion in LTE random access. Massive random access was studied in [23] using a learning automata based ACB scheme by estimating the traffic and adjusting barring factor. In [24], LSTM was used to predict number of preamble transmissions in each RAO, which was considered to be a time-series data. The authors in [25] proposed a reinforcement learning (RL) based access control to reduce the mean access delay. Different learning and non-learning based approaches for predicting traffic for access control optimization were surveyed in [26], where the authors pointed out that, as a future research direction learning-based approaches can be explored for priority-aware optimization in heterogeneous scenarios.

A recent work [27] proposed a QoS-based Dynamic and Adaptive Mechanism (QDAM) algorithm for heterogeneous scenario with different MTC devices having different delay requirements, where priority was given to delay-sensitive devices. QDAM uses the optimal barring rate p^*_{dacb} using the approach in D-ACB [19] as a baseline. Whenever the number of preamble collisions is more than a threshold, it does not allow any delay-tolerant devices whereas it allows delay-sensitive devices with probability p^*_{dacb}. On the other hand, if collision is less than the threshold, QDAM allows access to all the delay-sensitive devices and assigns barring rate of p^*_{dacb} for delay-tolerant devices.

B. Motivation and key contributions

ACB distributes the incoming traffic such that the maximum possible number of devices get to access the eNB when they ask for it. However, in the process, access contention introduces additional delay. In a heterogeneous scenario like in mMTC, where different devices have widely-varying QoS requirements, different groups of MTC devices serving different applications may have their own latency budgets and hence unique delay sensitivities. Smart grid communication is one such use case, where different applications have widely varying delay budgets ranging from 100 ms to 5 s [28]. Health-related IoT applications are also time-constrained [2]. In those applications, packets may be dropped if the random access delay crosses the delay budgets of the respective applications.

It is notable that, although there are multiple ACB algorithms proposed in the literature, most of them did not consider delay-constrained scenarios and are not truly delay-aware. Also, in the related works, all MTC devices are either considered to be in the same access class (AC) [19], or even if they are differentiated, their priorities are decided beforehand [21]. The approach in D-ACB [19] did not consider the delay sensitivity to adjust barring rates of devices. Although QDAM [27] considered delay-sensitive devices, it also keeps the priority of devices fixed. Thus, in the existing approaches, there is no mechanism to dynamically prioritize the MTC devices in terms of some performance metrics, such as the respective delay budgets. We argue that, like dynamic routing for real-time QoS support [29], dynamically accounting for the packet access delay of the contending nodes and updating priority of devices in ACB is expected to improve the preamble transmission success rate and hence QoS support. To this end, in this paper a novel delay-aware priority access classification (DPAC) based ACB is proposed. The major contributions of the paper and significance are as follows:

1) In the proposed DPAC for delay-sensitive mMTC applications, access priority of the devices are dynamically assigned based on the packet delay due to ACB.
2) A parametric model is proposed to assign barring rate corresponding to each priority access class. Optimal values of the model parameters of DPAC are determined by solving an optimization problem at the eNB that maximizes the overall system utility.
3) A lower bound of the maximum achievable utility is found analytically by its polynomial approximation and solving the dual of the objective function.
4) Further, a RL aided framework, named DPAC-RL, is proposed which uses a RL agent to update the model parameters of DPAC in an online learning setup by taking feedback from the dynamic environment.
5) Performance of the proposed DPAC and DPAC-RL based ACB schemes are compared with the existing static ACB, barring bitmap based EAB [13], D-ACB [19], and QDAM [27] algorithms via extensive simulations.
6) In homogeneous scenario with only delay-sensitive MTC devices, compared to the closest D-ACB algorithm, DPAC offers up to $\sim 40\%$ gain in preamble transmission success rate, whereas DPAC-RL achieves up to 75% improvement. In heterogeneous scenario with both delay-sensitive and delay-tolerant MTC devices, DPAC-RL is able to gain up to $\sim$ 60% compared to the closest

Table I: RB available for different bandwidth

Available bandwidth (MHz)	1.4	5	10	20
Total RB	6	25	50	100

Table II: Standard access classes in ACB

AC	**Type**
$0-9$	Normal UE
10	Emergency Calls
$11-15$	Higher priority services (PLMN, Security, Public Utilities, Emergency, PLMN Staff)

competitive scheme QDAM.

C. Paper organization

The layout of the paper is as follows. In Section II, an overview of the LTE random access and the proposed system model are presented. Section III contains the proposed DPAC algorithm and related analysis. System utility optimization to find the optimal model parameters is presented in Section IV. Section V introduces a RL-aided framework (DPAC-RL) of the proposed scheme. Performance results are discussed in Section VI, followed by the concluding remarks in Section VII.

II. System model

In this section, first a brief overview of contention-based random access in LTE from mMTC context is given. Then, the proposed modification in the protocol architecture is outlined.

A. Contention-based random access in LTE

At the beginning of each RAO, the eNB broadcasts all basic configuration parameters through SystemInformationBlock-Type2 (SIB2). The MTC device (a type of user equipment (UE)) chooses one of the available preambles randomly and sends it to the eNB over physical random access channel (PRACH). In contention-based random access, 54 such preambles are available to one eNB. Preambles are orthogonal signatures generated using Zadoff-Chu sequences [30].

A single resource block (RB, the minimum unit of resource in LTE) is constituted by one sub-frame (1 ms) and twelve sub-carriers ($12 \times 15 = 180$ kHz). The number of RBs in each RAO depends on the available bandwidth, as shown in Table I. The limit on maximum number of successive preamble transmission attempts of an MTC device is decided by SIB2 parameter *preambleTransMax*. If more than one MTC devices choose the same preamble and simultaneously transmit them to the eNB, it leads to a collision, which is discussed in below. Once the preamble is successfully detected, in the second step, eNB sends random access response (RAR) message to the MTC device which includes information related to the uplink transmission, namely, timing alignment, uplink resources reserved for the MTC device, and an identifier. After the RAR is received, the MTC device does the necessary time synchronization using the received time correction. In step three, the MTC device transmits L2/L3 message (RRC connection request) to eNB, which also includes its identity. Lastly, in step four, eNB sends a contention resolution message to all those MTC devices with packets successfully decoded.

Due to orthogonality of preambles, different MTC devices can access an eNB simultaneously in a RAO using different preambles. As the MTC devices choose preambles randomly, there may be a scenario when multiple MTC devices trying to connect to the same eNB choose the same preamble in a random access slot. This event may lead to two possibilities: the collision is detected either in Step 1 or in Step 3 of random access procedure. Similar to what most researchers have considered [14], [21] and also as per what 3GPP specification suggests, the following assumptions are taken: while sending a preamble to the eNB, MTC devices do not send any identifier. If multiple MTC devices ask for the same preamble to connect to an eNB in the same RAO, it goes undetected in Step 1. Consequently, the eNB provides them with the same RB to transmit random access data in Step 3 which may lead to collision. When the packet transmission succeeds, eNB sends a contention resolution message to MTC device indicating successful detection. If the preamble transmission fails due to collision, the MTC device waits for a random back-off time $t_{bo} \sim U(0, b_i)$, where the back-off indicator, b_i can take any value in between 0 and 960 ms and is decided by the eNB.

B. Overload / Congestion control

In MTC, while the amount of data to be sent is very less, the number of devices simultaneously accessing a single eNB can be much higher than that in conventional H2H communication. The frequency at which these MTC devices try to send data is also much higher than that in H2H scenario. However, the number of preambles is limited; 54 for contention-based random access, which are shared by M2M and H2H communications [31]. As suggested in [9] and [19], MTC and H2H devices are considered to choose from separate sets of preambles. The devices suffering from collision re-attempt for access at a later RAO, thus adding to the load. Also, there may be scenarios where one event may trigger different types of MTC devices, which may worsen the congestion. Therefore, in mMTC, with the increasing incoming traffic, the LTE random access can be unstable, which necessitates the requirement of an efficient congestion/overload control technique.

C. Access class barring in mMTC

Among the 3GPP recommended solutions for congestion control, ACB is one of the most effective techniques [9]. All UEs are assigned one of the 16 access classes AC0 to AC15 which is stored in the UE's Subscriber Identity Module (SIM/USIM). Different UEs are grouped together in different access classes (ACs) based on their specific applications. The standard set of ACs suggested by 3GPP are shown in Table II. In the existing convention, an MTC device is considered as normal UE and hence it is assigned one of the classes from ACs 0-9 [10]. At the beginning of each RAO, eNB broadcasts the ACB parameters: barring rate $p_{acb} \in \{0.05, 0.1, \cdots, 0.3, 0.4, \cdots, 0.7, 0.75, 0.8, \cdots, 0.95\}$, and mean barring time $t_{acb} \in \{4, 8, 16, \cdots, 512\}$ (s) of every AC through SIB2 [10]. The MTC devices wanting to send data to the eNB randomly generate a uniformly distributed random number r_u between 0 and 1. If r_u is less than p_{acb}, then

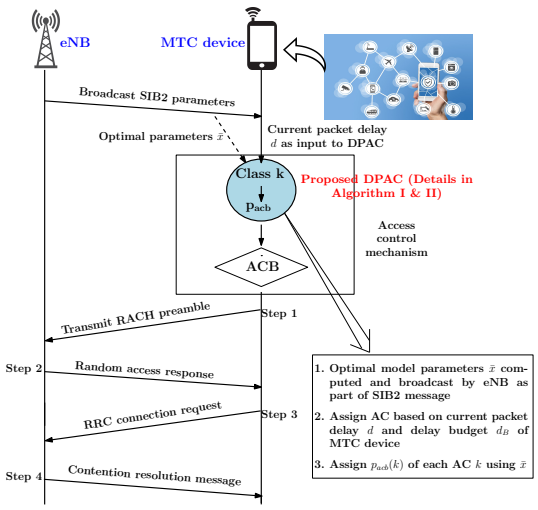


Figure 1: Random access with proposed DPAC-based ACB.

it clears the ACB and gets to contend in the current random access phase. Else, it has to wait for a random back-off period of time [7], which is calculated as $t_{acb}(0.7 + 0.6r_{u_2})$, where r_{u_2} is a uniform random number in $[0, 1)$. Every t_{rao} subframe has a RAO and each subframe is of duration $t_{sf} = 1$ ms. To support mMTC along with H2H, 3GPP suggests separate (one or more) AC(s) may be used for MTC devices [9].

D. Proposed modification for delay-sensitive scenarios

In conventional ACB, the access classes are hard-coded in the devices, usually in the SIM/USIM. Hence their access priorities are fixed beforehand and never change throughout the random access procedure. However, there may be scenarios where the same MTC device may be barred multiple times in consequent random access attempts, and hence it encounters higher delay. In delay-constrained applications, these devices must be given higher priority in access contention, and hence the standard ACB technique needs to be modified. To this end, in the proposed DPAC-based ACB scheme, priority classes of MTC devices are decided dynamically based on their current delay. Thus, the same MTC device may be put in different access classes dynamically in consecutive RAOs based on the waiting delay encountered. As shown in Fig. 1, the proposed scheme modifies the access control stage before random access procedure, by introducing the DPAC strategy.

III. Analysis of Proposed DPAC-based ACB

In this section, the proposed delay-aware access prioritization in ACB is presented in detail, along with the related analysis and performance characterization.

A. Proposed priority access classification (DPAC)

The proposed DPAC scheme dynamically assigns barring rate p_{acb} to the MTC devices in two steps. Firstly, it assigns priority class to an MTC device based on the delay criticality of the HoL packet in its transmit buffer. If total K classes are available, d_B is the delay budget, and d is the delay encountered by the MTC device due to ACB, then its access class is assigned according to the following rule:

$$\text{If } d_B\frac{k-1}{K} < d \leq d_B\frac{k}{K}, \text{ then } AC = k, \forall k \in \{1, \cdots, K\}.$$

Subsequently, barring rate is assigned for each access class. It might be noted that p_{acb}, which is conventionally called barring rate, is actually the success probability of ACB. A higher value of p_{acb} (i.e., a higher access priority) is assigned for the classes of devices which are on the verge of crossing their respective delay budgets. Thus, the barring rate $p_{acb}(k)$ of access class k is an increasing function of k. The barring rate of class k device is proposed as:

$$p_{acb}(k) \triangleq p_k = x_1 k^{x_2} + x_3. \tag{1}$$

Optimal values of the parameters x_1, x_2, x_3 are obtained via an optimization problem formulation as presented in Section IV. The optimal parameters x_1, x_2, x_3 are broadcast as part of the SIB2 message by eNB at the beginning of each RAO slot. The MTC device asking for access in a RAO, assigns the corresponding AC based on its current delay and the priority-based AC assignment logic available with it. Subsequently, it can compute its barring rate p_{acb} using the optimal parameters received from eNB and the AC assigned to it. Once the incoming MTC device is assigned an AC and the corresponding probability, the conventional ACB mechanism follows. If the packet delay of any MTC device crosses its allowed delay budget, the packet is dropped, and that MTC device does not take part in ACB contention in that RAO. The flow of the proposed DPAC-based ACB technique is presented in Algorithm 1.

B. Arrival and delay characterization

The proposed DPAC-based ACB scheme allows a fraction of the contending MTC devices. Therefore, it is required to model the number of devices contending for access in each RAO (both new arrivals and backlogged ones from previous RAOs). In mMTC, a huge number of devices can be activated simultaneously within a very short activation interval τ_A. Beta

Algorithm 1: Proposed DPAC-based ACB

1 At the beginning of a RAO, eNB broadcasts barring time t_{acb}, total number of allowed classes K, and the optimal parameter set x_1, x_2, x_3;
2 *for each MTC device* **repeat**
3 Assign priority class based on delay d of MTC device and its delay budget d_B: **if** $d_B\frac{k-1}{K} < d \leq d_B\frac{k}{K}$ **then** AC = k, $\forall k \in \{1, \cdots, K\}$;
4 Each device in class k is assigned access probability $p_{acb}(k) = x_1 k^{x_2} + x_3$;
5 Generate a uniform random number $r_u \sim U[0, 1)$;
6 **if** $r_u \leqslant p_{acb}$ **then** Random Access is initialized;
7 **else** Generate new random number $r_{u_2} \sim U[0, 1)$;
8 Waiting time calculated as $t_{acb}(0.7 + 0.6r_{u_2})$;
9 **until** *Random Access for the MTC device is initialized*;

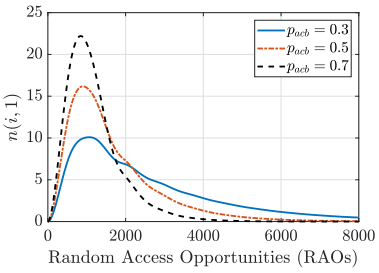


Figure 2: Number of new MTC access requests in i^{th} RAO.

distribution is considered as the most appropriate to characterize the bursty arrivals in mMTC [9]. In beta distributed traffic model, activation time of the MTC devices follows a scaled beta distribution in $[0,\ \tau_A]$, which is given by:

$$f_A(i) = \frac{i^{\alpha-1}(\tau_A - i)^{\beta-1}}{{\tau_A}^{\alpha+\beta-1}\mathcal{B}(\alpha,\beta)},\ \ 0 \le i \le \tau_A \tag{2}$$

$$\text{where}\ \ \mathcal{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\ \text{and,}\ \Gamma(n) = (n-1)!.$$

Traffic model 2 in [9] is used in this work. It follows Beta$(3, 4)$ distribution over 10 s duration, which accounts for τ_A RAOs.

$$f_A(i) = \frac{60i^2(\tau_A - i)^3}{{\tau_A}^6},\ \ 0 \le i \le \tau_A. \tag{3}$$

The total number of access requests in RAO-i includes the new arrivals in that RAO and the attempts from the backlogged devices. Random variable (RV) A denotes the RAO in which an MTC device is active for the first time, and the RV D denotes the delay due to ACB in terms of number of RAOs. If $n(i, pt)$ denotes the number of MTC devices attempting pt^{th} preamble transmission, then it can be expressed as:

$$n(i,1) = N \cdot \Pr(A + D = i) = N \sum_{q=0}^{i} f_A(q) f_D(i-q) \tag{4}$$

where N is the total number of MTC devices having preamble transmission requests over the duration of τ_A RAOs and f_D is the probability density function (PDF) of D, computed using a recursive technique, as shown in Appendix A.

The number of preamble re-transmission attempts $n(i, pt)$ $\forall pt \in \{2, 3, \cdots, preambleTransMax\}$ is found using recursion [17], starting from (4), as derived in Appendix B. Subsequently, the total number of MTC devices contending for access in i^{th} slot is given by:

$$n(i) = \sum_{pt=1}^{preambleTransMax} n(i, pt). \tag{5}$$

The number of devices contending for preamble transmission for the first time in different RAOs is shown in Fig. 2.

C. Poisson binomial distribution and binomial approximation

Here, the MTC devices arrival modeling in Section III-B is used for further analysis of the proposed DPAC-based ACB. Maximization of the number of successful transmissions in each slot will lead to minimization of total service time.

Let the RVs Λ_i and Λ_i^p respectively denote the number of MTC devices attempting access and the number of devices passing the ACB check in the i^{th} RAO. If Υ_i denotes the total number of successful preamble transmissions in i^{th} RAO, then the expected number of successful transmissions is given by:

$$E[\Upsilon_i|\Lambda_i = n] = \sum_{j=1}^{n} E[\Upsilon_i|\Lambda_i^p = j] P(\Lambda_i^p = j|\Lambda_i = n). \tag{6}$$

If total N_{pr} preambles are available, only those preambles which are chosen by only one MTC device will result in successful transmission. Let Ψ_q be the RV denoting the number of MTC devices choosing the q^{th} preamble. Then,

$$\begin{aligned} E[\Upsilon_i|\Lambda_i^p = j] &= \sum_{q=1}^{N_{pr}} P(\Psi_q = 1|\Lambda_i^p = j) \\ &= N_{pr}\binom{j}{1}\frac{1}{N_{pr}}\left(1 - \frac{1}{N_{pr}}\right)^{j-1}. \end{aligned} \tag{7}$$

If all the MTC devices have the same barring rate p_{acb}, then it results in a binomial distribution constituted by independent and identically distributed (i.i.d.) Bernoulli trials, given by:

$$P(\Lambda_i^p = j|\Lambda_i = n) = \binom{n}{j} {p_{acb}}^j (1 - p_{acb})^{n-j}. \tag{8}$$

In contrast, in the proposed DPAC-based ACB, the m^{th} MTC device is assigned barring rate p_m, based on its delay encountered. Hence the individual Bernoulli trials are not necessarily identical here. This results in Poisson binomial distribution [32] which is characterized by:

$$P(\Lambda_i^p = j|\Lambda_i = n) = \sum_{S \in F_j} \prod_{s \in S, |S|=j} p_s \prod_{u \in U, |U|=n-j} (1 - p_u). \tag{9}$$

S and U are respectively the sets of successful and unsuccessful MTC devices, F_j is the set of all subsets of j integers that can be selected from $\{1, 2, 3, \cdots, n\}$, and p_s and p_u are respectively the barring rates of successful and unsuccessful MTC devices. $U = S^c = \{1, 2, \cdots, n\} - S$. A closed-form expression for the probability in (9) is obtained as [32]:

$$\Pr(\Lambda_i^p = j|\Lambda_i = n) = \frac{1}{n+1}\sum_{l=0}^{n} C^{-lj} \prod_{m=1}^{n} \left(1 + (C^l - 1)p_m\right) \tag{10}$$

where $C = \exp\left(\frac{2i\pi}{n+1}\right)$ and $i = \sqrt{-1}$.

In the proposed model, the values of p_m for all $m \in \{1, 2, \cdots, n\}$ are not different. Instead, all the devices belonging to class k are assigned the same probability p_k. Thus, if L_k denote the number of devices assigned class k then,

$$\begin{aligned} \Pr(\Lambda_i^p = j|\Lambda_i = n) &= \sum_{l=0}^{n} \frac{C^{-lj}}{n+1} \prod_{k=1}^{K} \left(1 + (C^l - 1)p_k\right)^{L_k} \\ &= \frac{1}{n+1}\sum_{l=0}^{n} C^{-lj} x_l = \frac{1}{n+1} X_j \end{aligned} \tag{11}$$

where $x_l = \prod_{k=1}^{K} \left(1 + (C^l - 1)p_k\right)^{L_k}$ and $\{X_j\} = DFT\{x_l\}$.

Due to mathematical intractability of the Poisson binomial distribution, it is approximated as a binomial distribution

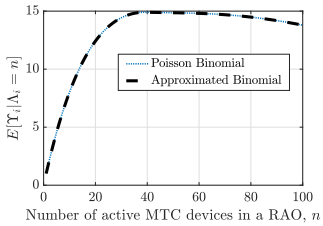


Figure 3: Binomial approximation of Poisson binomial distribution.

[33] in the rest of the paper. The success probability $\bar{p}$ of the approximated binomial distribution is the average success probability of all the n active devices, given by:

$$\bar{p} = \frac{1}{n}\sum_{m=1}^{n} p_m = \frac{1}{n}\sum_{k=1}^{K} L_k p_k = \frac{1}{n}\sum_{k=1}^{K}(nc_k)(x_1 k^{x_2} + x_3) = \sum_{k=1}^{K} c_k(x_1 k^{x_2} + x_3) \quad (12)$$

where c_k is the probability that a device belongs to class k. Hence, probability of j devices getting through ACB filtering among n active MTC devices can be expressed as:

$$P(\Lambda_i^p = j|\Lambda_i = n) = \binom{n}{j}\bar{p}^j(1-\bar{p})^{n-j}. \quad (13)$$

Using (7) and (13) in (6), we have the expected number as:

$$E[\Upsilon_i|\Lambda_i = n] = n\bar{p}\left(1 - \frac{\bar{p}}{N_{pr}}\right)^{n-1}. \quad (14)$$

Fig. 3 by compares the values of $E[\Upsilon_i|\Lambda_i = n]$ from the actual and the approximate distributions at different values of n, showing that the binomial distribution with success probability $\bar{p}$ is able to approximate the equivalent Poisson binomial distribution very well. Also, it was observed in [33] that the approximation of Poisson binomial distribution with binomial distribution works well if the ratio of variances of the actual and the approximated distributions tends to 1.0. In the current context this ratio is 0.9968, which is close to 1.0.

The analysis and the expressions derived in this section are used in the following section to maximize the system utility.

IV. Optimization of model parameters

In this section an optimization problem is formulated to obtain the model parameters that maximize the system utility. These parameters are computed by the eNB at the start of each RAO and broadcast to the MTC devices. The active devices then determine their respective barring rates for their access.

A. Problem formulation

The utility in the proposed setup is a combination of throughput maximization and packet drop probability minimization from the devices due to violation of the respective delay budget d_B. The throughput $\mathbb{T}$, defined as the average rate of successful preamble transmissions (where the number of arrivals in each slot $n \leq n_{max}$), is defined as:

$$\mathbb{T} = \sum_{n=1}^{n_{max}} E[\Upsilon_i|\Lambda_i = n] \cdot \Pr[\Lambda_i = n] = \sum_{n=1}^{n_{max}} n\bar{p}\left(1 - \frac{\bar{p}}{N_{pr}}\right)^{n-1} \Pr[\Lambda_i = n]. \quad (15)$$

The corresponding packet drop probability $\mathbb{P}_{cross}$, i.e., the probability that the access delay $d > d_B$, is defined as:

$$\mathbb{P}_{cross} = \Pr(d > d_B) = 1 - F_D(d_B) \quad (16)$$

where F_D is the cumulative distribution function (CDF) of the delay due to ACB, modeled using a recursive algorithm [17] (see Appendix A). The probability of n devices being active in i^{th} RAO, i.e., $\Pr[\Lambda_i = n]$, is modeled using the expression of number of arrivals in each RAO, given by (5).

A multi-objective optimization problem is formulated and solved using weighted sum method. Combining the throughput and delay factors with equal weights, the formulation is:

$$\begin{aligned}(P) : &\underset{\bar{x}}{\text{minimize}}\, f_0 = -\widetilde{\mathbb{T}} + \mathbb{P}_{cross} \\ \text{s.t. } &C_1 : 0 \leq p_k \leq 1, \forall k \in \{1, 2, \cdots, K\} \\ &C_2 : N_{pr} \leq n\bar{p} \leq n \end{aligned} \quad (17)$$

where the optimization variable is the vector: $\bar{x} = [x_1, x_2, x_3]$, $\widetilde{\mathbb{T}}$ is the normalized value of the actual throughput $\mathbb{T}$, f_0 is the composite objective function, p_k and $\bar{p}$ are given by (1) and (12), respectively, while c_k can be obtained as:

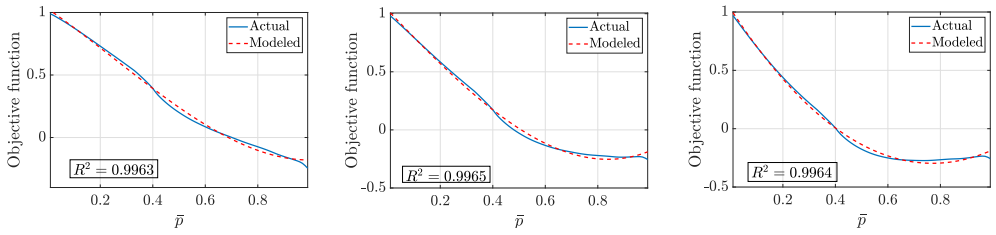
$$c_k = \Pr\left(d_B\frac{k-1}{K} < d \leq d_B\frac{k}{K}\right) = F_D\left(d_B\frac{k}{K}\right) - F_D\left(d_B\frac{k-1}{K}\right) \quad (18)$$

The first set of constraints C_1 corresponds to p_k, which is a probability in $[0, 1]$. C_2 makes sure that the preambles are not under-utilized. If n MTC devices ask for access in one RAO and $n \geqslant N_{pr}$, then among them the number of devices allowed by ACB should not be less than the number of preambles N_{pr}. It is notable that, if $n < N_{pr}$, the barring mechanism is not enabled, i.e., $p_k = 1$ for all classes. Since $\mathbb{P}_{cross}$ is a probability, it is in $[0, 1]$. To make sure that the other term in the objective function also lies within this range, $\mathbb{T}$ is normalized to $\widetilde{\mathbb{T}}$. In multi-objective optimization, each of the individual objective functions is normalized by their respective optima. Therefore, in this work normalizing factor of $\mathbb{T}$ is computed by using the expression for the maximum possible expected number of preamble transmission for a given n active devices, which is obtained in Lemma 1.

Lemma 1. *The maximum value of the expected number of successful preamble transmissions $E[\Upsilon_i|\Lambda_i = n]$ for a given number of active MTC devices n is upper bounded by $\frac{nN_{pr}}{n-2}\left(1 - \frac{1}{n-2}\right)^{n-1}$, where N_{pr} is the number of available preambles.*

Proof. See Appendix C. □

Closed-form expression cannot be achieved for the objective function f_0 as the CDF F_D is computed using a recursive function. For a lower bound of f_0, a polynomial approximation is used, as presented in the following subsections.



(a) Delay budget, $d_B = 1$ s (b) Delay budget, $d_B = 5$ s (c) Delay budget, $d_B = 10$ s

Figure 4: Polynomial (cubic) approximation of the objective function at different delay budgets. $N = 30000$ and $N_{pr} = 40$.

It may be noted that the delay PMF in [17] models the delay encountered by the devices at the end of the observation window. Hence, the utility maximization, which includes throughput optimization as well, should also be done over the whole period, not for a single RAO. To take care of this, the optimization framework is made arrival-based, i.e., the optimization variables or the model parameters are tuned based on the arrival in a RAO. In that case, the system needs to know the arrival distribution of different RAOs beforehand, so that it can optimize the parameters based on that. How the primal problem is quasi-convex and how it is solved is explained in Appendix D as part of the proof of Lemma 2.

Lemma 2. *The composite objective function f_0 is a quasi-convex function in $\bar{x}$.*

Proof. See Appendix D. □

The optimal model parameters which are obtained by solving the primal problem (17) are used to evaluate the performance of the proposed algorithm in Section VI.

B. Polynomial approximation

As explained in the previous section, due to recursive techniques being used, a closed-form expression cannot be achieved for the composite objective function f_0. For some analytical insight on the optimization problem, its polynomial approximation is used. It is observed that, for all values of delay budget d_B, f_0 can be approximated by a cubic function of $\bar{p}$, given as: $f_0 \approx \alpha\bar{p}^3 + \beta\bar{p}^2 + \gamma\bar{p} + \delta$, where the parameters $\alpha, \beta, \gamma, \delta$ are obtained by curve fitting. From Fig. 4 it can be noted that the approximation is reasonably good, with R^2 value of the fit more than 0.99 in all cases. The polynomial fit is shown in Fig. 4 for different delay budgets with $N = 30000$ and $N_{pr} = 40$. Similar fits have been observed for other values of N and N_{pr} as well, which are not shown here.

Lemma 3. *The composite objective function f_0 can be approximated by a cubic polynomial function in $\bar{p}$. The optimal value of the primal problem, p^* is lower bounded by:*

$$d^* = \alpha\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right)^3 + \beta\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right)^2 + \gamma\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right) + \delta$$

where $\alpha, \beta, \gamma, \delta$ are the parameters of the cubic model.

Proof. See Appendix E. □

The lower bound of f_0, as computed in Lemma 3, shows that the best performance achievable by DPAC depends on the model parameters of the cubic polynomial approximation.

The effect of utility maximization achieved using the optimization framework is analyzed in the Section VI.

V. RL Aided Framework (DPAC-RL)

The DPAC framework proposed and optimized in Sections III and IV may give sub-optimal output because of dynamic nature of the system. For example, as the delay distribution changes, the assigned p_{acb} is updated, leading to a change in the expected number of devices arriving in each slot, which in turn changes the delay distribution. In such scenarios RL is usually preferred because of its ability to collect feedback from the past experiences and act accordingly. We consider a RL agent at the eNB that decides the optimal actions, i.e., optimal values of the set of model parameters x_1, x_2, x_3 to maximize the percentage of successful preamble transmissions in each RAO, based on the values of some state variables.

A. Modeling as a RL problem

The basic entities of any RL framework are defined here as: State $\mathfrak{s} \in \mathbb{S}$, Action $\mathfrak{a} \in \mathbb{A}$ and Reward $\mathfrak{r} \in \mathbb{R}$, where $\mathbb{S}, \mathbb{A}$ and $\mathbb{R}$ are the corresponding sets they are chosen from. The state vector at the i^{th} RAO, s_i comprises of the following observations from the previous RAO: number of active MTC devices $n^{(i-1)}$, number of collided preambles $m_c^{(i-1)}$, number of MTC devices which passed in ACB $n_p^{(i-1)}$, number of unused preambles $m_u^{(i-1)}$, number of MTC devices which successfully transmitted preamble $n_{succ}^{(i-1)}$, and number of devices dropped due to exceeding delay budget $n_d^{(i-1)}$. The action taken at the i^{th} RAO, denoted by a_i, corresponds to a triplet that consists of the values chosen for each of the three model parameters of DPAC: x_1, x_2, and x_3. For the sake of simplicity a discrete action space is considered. Values of the three model parameters x_1, x_2, x_3 are chosen from three different sets $\mathbf{X_1}$,$\mathbf{X_2}$, and $\mathbf{X_3}$, respectively. The reward $\mathfrak{r}$ gives an indication to the agent about how far the action taken in one step can contribute to the long-term goal of achieving the

system utility. The reward assigned to the agent on choosing action a_i, is given by:

$$r_{i+1} = \begin{cases} \dfrac{n_{succ}^{(i)}}{N_{pr}}, & \text{if } n^{(i)} \geqslant N_{pr} \\ \dfrac{n_{succ}^{(i)}}{n^{(i)}}, & \text{if } 0 < n^{(i)} < N_{pr} \end{cases} \tag{19}$$

where $n_{succ}^{(i)}$ denotes the total number of MTC devices which succeed in preamble transmission among $n^{(i)}$ active devices in the i^{th} RAO. It may be noted that, to make sure that constraints C_1 and C_2 in (17) are satisfied, very high negative reward is assigned for violation of either of them.

Based on the feedback from the environment in the form of state variables, the agent has to take decision, i.e. chose from the available set of actions. This is characterized by the policy function $\pi(\mathfrak{s}) = \mathfrak{a}$. To find the optimal policy, an agent has to quantify how good it is to take action $\mathfrak{a}$ at state $\mathfrak{s}$, which is done through state-action value function $q(\mathfrak{s}, \mathfrak{a})$. The optimal policy is learned using Q-Learning algorithm which is one of the most popular RL techniques.

B. Q-Learning

To estimate the value function for a given policy π, Q-Learning is used, which is an off-policy temporal difference (TD) control algorithm [34]. TD methods are advantageous compared to Monte-Carlo or dynamic programming (DP) methods because of their faster convergence and model-free approach. Also, Q-Learning is preferred over on-policy algorithms like SARSA because its output is independent of the policy being followed.

In Q-Learning, the action-value function Q is learned according to the Bellman optimality equation, by the following update equation in i^{th} RAO [34]:

$$Q(s_i, a_i) \leftarrow Q(s_i, a_i) + \eta[r_{i+1} + \gamma_q \max_a Q(s_{i+1}, a) - Q(s_i, a_i)] \tag{20}$$

where η is the learning rate and γ_q denotes the discount factor.

In the basic tabular Q method, a table of Q values for each state action pair $(\mathfrak{s}, \mathfrak{a})$ is maintained, which is updated in each step following (20). But in complex scenarios with very large size of state space, tabular Q method is not scalable, leading to the need of a function approximator, which is discussed next.

C. Deep Q Network (DQN)

Although Q-Learning helps to get rid of the model dependence, it suffers from the curse of dimensionality due to large state space in real-life problems [35]. Consequently, it might be difficult to achieve the optimal policy with time and resource constraints. Hence, a suitable function approximator is required for the approximation of value function for state-action pairs. Deep Neural Networks (DNN) are well reputed in different applications to approximate complex functions even in large dimensional scenarios. A DNN used to model the state-action value function of Q-Learning is called a Deep Q Network (DQN). We consider the basic fully-connected architecture for the DNN, which takes state as the input and outputs the action values. As the activation function, rectified linear unit (ReLU), given by $f_{ReLU}(x) = \max(0, x)$, is used.

Algorithm 2: Proposed DPAC-RL framework

Input: DQN related hyper-parameters, barring time t_{acb}, total number of allowed classes K

Local Q network initialized with weights $\mathbf{w}$;
Target Q network initialized with weights $\mathbf{w}^- = \mathbf{w}$;
Replay buffer is initialized to its full capacity B_R;
for *each episode* **do**
  State vector is initialized as $\{0, 0, 0, N_{pr}, 0, 0\}$;
  for *each RAO slot* **do**
    Predict action values from current state;
    With probability ϵ randomly choose an action, otherwise, with probability $1 - \epsilon$ choose action corresponding to maximum action value;
    Broadcast t_{acb}, K, and $\{x_1, x_2, x_3\}$ (as per the action chosen in Step 8);
    for *each active MTC device* **do**
      Steps $3 - 8$ of DPAC based ACB (Algorithm 1)
    Save experience $< s_i, a_i, r_{i+1}, s_{i+1} >$ in replay buffer;
    if *at least L_M samples available in the replay buffer* **then** Randomly sample mini-batch of experiences ;
    Minimize loss function in (21) using ADAM optimizer to update weight $\mathbf{w}$ of local network;
    Once in every U_T steps, copy updated weights of local network $\mathbf{w}$ to weights $\mathbf{w}^-$ of target network;

In each epoch or RAO slot, given a state, the agent uses the DQN to predict the Q values corresponding to each of the actions. The action corresponding to the highest Q value is chosen by the agent. In the learning phase, the DQN model computes the loss, i.e. the mean squared error between the actual and the target Q values. The mean squared error (MSE) loss function at the j^{th} training iteration is given by:

$$L_j(\mathbf{w}_j) = \mathbb{E}_{s_i, a_i, r_{i+1}, s_{i+1}}[(r_{i+1} + \gamma_q \max_a Q(s_{i+1}, a; \mathbf{w}_j) - Q(s_i, a_i; \mathbf{w}_j))^2] \tag{21}$$

Subsequently, this error is back-propagated to update the weights of the model such that the loss is minimized. To update parameter $\mathbf{w}$ of the DQN approximator for loss minimization, ADAM optimizer [36] is used, which combines the advantages of the adaptive gradient (AdaGrad) and root means square propagation (RMSProp) algorithms. Flow of the proposed DPAC-RL framework is shown in Algorithm 2.

In order to maximize reward the agent chooses the action corresponding to the maximum Q value. But if it always does so, it might miss out some actions which could have produced higher reward but were never tried. To address this issue, a balance between exploitation (choose an action that is known to give best result) and exploration (choose a random action

Table III: DQN hyper-parameters

Hyper-parameters	**Values**
Learning rate η	5×10^{-4}
Discount factor γ_q	0.5
Mini-batch size L_M	64
Replay buffer size B_R	10^5
Exploration probability ϵ	1 to 0.01
Decay rate of ϵ	0.995
Target network update frequency U_T	4

which may help to choose better actions in future) is practiced. In this work, $\epsilon-$greedy approach [37] is used to maintain this trade-off, where the agent chooses a random action with probability ϵ and selects the action corresponding to highest Q value with probability $1-\epsilon$.

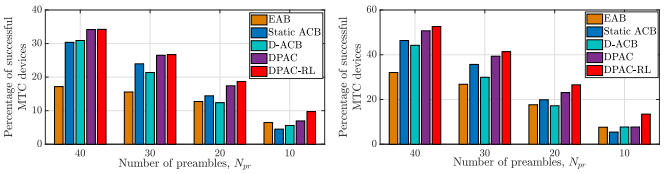
A basic assumption of most of the machine learning algorithms like neural networks is that the training data is independent and identically distributed (i.i.d.). If the model is trained with sequentially correlated data, which is very much possible in real-life scenarios, it may affect the convergence and performance of the DQN. To combat this we use the concept of experience replay [38], where training data is randomly sampled from a large buffer of past experiences. The expectation in (21) is done over a mini-batch of previous samples randomly chosen from the replay buffer. RL algorithms may also suffer from policy oscillation due to the fact that even a small change in Q may lead to significant change in the policy. To counter this instability issue, the concept of Q-Learning update with target networks is used [38]. The weights of the target Q network $\mathbf{w}^-$ are updated once in every U_T steps by copying weights of the local Q network $\mathbf{w}$. The MSE loss function at the j^{th} training iteration with target networks is given by:

$$L_j(\mathbf{w}_j) = \mathbb{E}_{s_i,a_i,r_{i+1},s_{i+1}}[(r_{i+1}+\gamma_q \max_a Q(s_{i+1},a;\mathbf{w}_j^-) - Q(s_i,a_i;\mathbf{w}_j))^2]. \quad (22)$$

Performance of the proposed DPAC-RL framework along with that of the proposed DPAC based ACB are evaluated through extensive simulation in the following section.

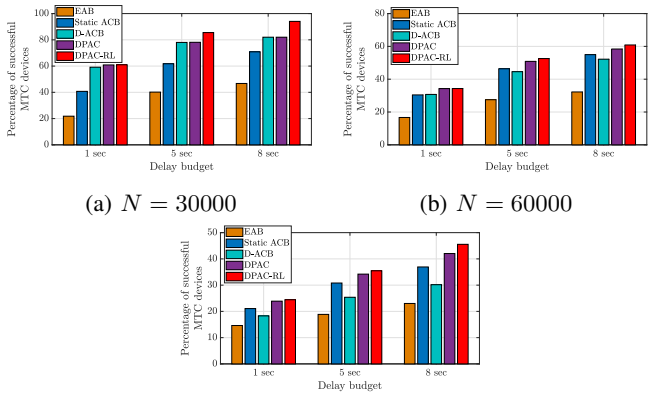
VI. Performance evaluation

In this section the performance of DPAC-based ACB is evaluated in terms of the key performance indicators (KPIs) that are described in the following subsection. Next, the DPAC-based ACB is compared with the existing static ACB, EAB [13], and D-ACB scheme [19], to find the achievable gain by the DPAC. Note that the output of the optimization framework is the set of optimal model parameters for each possible number of arrivals. In a real system, the number of arrivals n in a RAO is not exactly known. Instead, the eNB will have to estimate it in each RAO. For fair comparison, the same estimation algorithm is used as in [19] to estimate n. The system parameter values are: $preambleTransMax = 10$, $K = 6$, $t_{acb} = 4$ s, $t_{rao} = 5$ sub-frames $= 5$ ms, $b_i = 20$ms. Hence, the beta distribution interval of 10 s is equivalent to $\tau_A = 2000$ RAOs. It is also notable that, following the existing literature, fixed barring rate p_{acb} is kept at 0.5 for



(a) Delay budget = 1 s (b) Delay budget = 5 s

Figure 5: Successful devices versus N_{pr}. $N = 60000$.



(a) $N = 30000$ (b) $N = 60000$

(c) $N = 90000$

Figure 6: Successful MTC devices versus delay budgets. $N_{pr} = 40$.

the static ACB. DQN related hyper-parameters used in this performance evaluation are listed in Table III. The discrete values of the action elements are chosen from three different sets $\mathbf{X_1} = \{0.01, 0.02, 0.03, 0.05\}$, $\mathbf{X_2} = \{0.05, 0.1, 0.5, 1\}$, and $\mathbf{X_3} = \{0.1, 0.3, 0.5, 0.7\}$. The values in each of the sets are chosen such that the corresponding $p_{acb}(k)$ values are within 0 and 1 for all classes and also $p_{acb}(k)$ increases with k such that devices with higher priority are assigned higher success probability in ACB. The DQN is constituted of two hidden layers with 64 nodes in each of them. The RL agent is trained for multiple episodes till convergence, where each episode consists of 8000 RAOs which is same as the observation interval τ_O introduced in Section VI-A.

A. Key performance indicators (KPI)

The DPAC framework is used to reduce the access congestion in delay-sensitive mMTC. The following three key performance indicators (KPIs) capture the system performance:

1) *Success percentage:* The percentage of MTC devices that are able to successfully transmit the preambles in the observation period $\tau_O = 8000$ RAOs.
2) *Mean delay of served devices:* Mean ACB delay of the successful MTC devices that are successful in preamble transmission within the observation period.
3) *Drop percentage:* Fraction of devices having packets dropped because of the delay encountered in ACB exceeding their delay budget.

B. Homogeneous traffic scenario

In homogeneous MTC scenario, all the devices are considered delay-sensitive with the same delay budget.

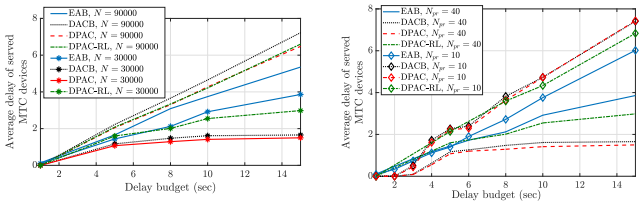
Success rate: In Fig. 5, percentage of successful preamble transmissions are compared with the competitive schemes for two different delay budgets 1 s and 5 s with total number of arrivals $N = 60000$. DPAC based ACB offers up to 40% more successful preamble transmission compared to the nearest competitive D-ACB; this gain is 49% compared to static ACB scheme. When the number of available preambles N_{pr} is 40, the proposed DPAC-RL framework is able to achieve up to 19% gain compared to D-ACB algorithm. As the number of available preambles is decreased, DPAC-RL is able to achieve up to 38%, 54% and 75% gain with $N_{pr} = 30, 20$ and 10, respectively, compared to D-ACB. Notably, as the number of preambles N_{pr} decreases, i.e., in more constrained scenario, the gain of the proposed algorithm increases.

It is observed that, percentage of successful preamble transmission in EAB is much lower; as low as one-third of DPAC-based ACB. This finding also corroborates the claims in [22] that in delay-sensitive MTC, ACB works better than EAB.

Effect of increasing total number of arrivals: As the number of preambles is limited, performance of ACB algorithms get affected by the number of arrivals in each RAO slot, which is again impacted by the total number of arrivals in an interval. DPAC performance at different total number of arrivals is shown in Fig. 6, for different delay budgets. It shows that, even when a higher number of preambles are available, e.g., with $N_{pr} = 40$, the gain of the proposed DPAC-based ACB in terms of throughput increases as the total number of MTC devices N increases. When $N = 30000$, the throughput of the proposed algorithm is about 3% higher than that with D-ACB. At $N = 60000$ and further at $N = 90000$, the gain of the proposed algorithm is increased to 14% and 39%, respectively. Fig. 6 also shows that the corresponding gains of DPAC-RL compared to D-ACB are 15%, 19% and 51%, respectively.

Delay performance: The mean delay of served devices with different delay budgets is shown in Fig. 7a with varying N, and in Fig. 7b with varying N_{pr}. The proposed DPAC-based ACB is observed to have lower delay performance compared to that of D-ACB. The reason is that, in DPAC as the encountered delay of an incoming MTC device increases, the system keeps on increasing its priority. The devices having delay bordering the budget are given the highest priority and hence they have the highest probability of success. This, in turn, brings down the average delay of the served devices. In the scenario where highest gain is achieved in terms of success percentage, i.e., in more constrained scenario with $N_{pr} = 10$, the delay of DPAC-based ACB is about 10% lower than that of D-ACB. It is observed that in order to achieve higher success percentage in preamble transmission, DPAC-RL encounters higher average delay compared to DPAC in some cases. However, it is not of much significance as still the delay is within the delay budget in all scenarios.

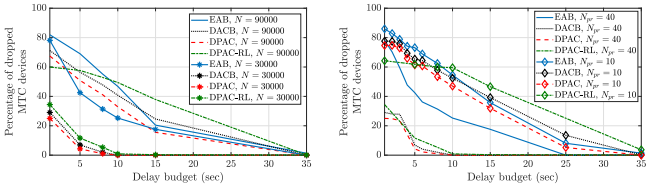
Packets dropped performance: MTC devices drop the packets that encounter delay greater than their respective delay budgets. The proposed DPAC algorithm gives higher priority to those MTC devices which are on the verge of crossing their respective delay budget. To quantify the benefit of DPAC, the drop percentage is also evaluated and the performance is compared in Fig 8. In Fig. 8a different values of N are considered while N_{pr} is fixed at 40. In Fig. 8b N_{pr} is varied while keeping N unaltered at 30000. Both the plots show decrease in percentage of dropped devices with increased delay budget. This is intuitive because, as budget is increased the system becomes more flexible in dropping packets. In all the scenarios drop percentage with the proposed DPAC is significantly lower, ranging 5 to 100% less compared to D-ACB. Fig. 8 also clearly shows that DPAC is most effective in reducing packet drop when delay budget is moderate.



(a) Number of total arrivals as parameter; $N_{pr} = 40$

(b) Number of preambles as parameter; $N = 30000$

Figure 7: Average delay of the served devices.



(a) Total number of arrivals as parameter; $N_{pr} = 40$

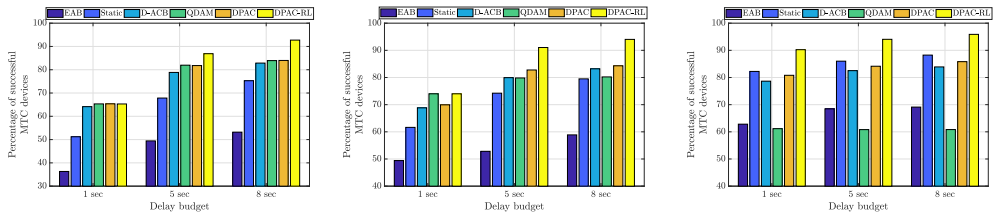
(b) Number of preambles as parameter; $N = 30000$

Figure 8: Unserved MTC devices due to exceeded delay budget.

As shown in Fig. 8a, when the delay budget is very high, e.g., 10 s in case of $N = 30000$, 15 s in case of $N = 60000$, and 25 s in case of $N = 90000$, the budget looses its significance. At such a high delay budget, the system tends to serve almost all packets, and the performance of the proposed technique in terms of drop rate converges with that of D-ACB. A similar trend is observed in Fig. 8b where N_{pr} is varied. The figures also show that percentage of devices dropped due to DPAC-RL is higher in most cases compared to DPAC based ACB. It is notable here that the dropped devices are also taken into consideration while calculating the percentage of successful preamble transmission in Figs. 5 and 6. That is, even after dropping more devices, overall percentage of successfully served devices is higher in DPAC-RL. Even if a device is not dropped due to exceeding delay budget, it can be unserved at later stage due to ACB or preamble collision. The target of the RL agent in DPAC-RL is to maximize the overall percentage of successful preamble transmission, irrespective of how many devices are dropped due to exceeding delay budget.

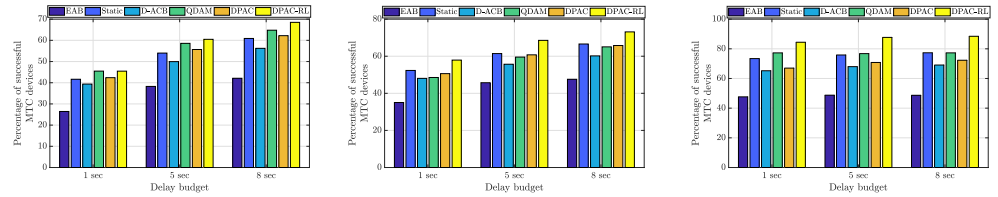
C. Heterogeneous traffic scenario

Unlike the homogeneous scenario where all nodes have the same delay budget, there may be scenarios where some of the MTC devices are more delay-tolerant than the others. In this section, system performance with such heterogeneous traffic is studied. Three levels of heterogeneity are considered where



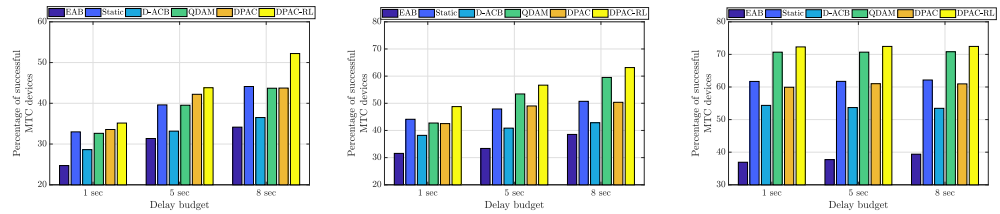
(a) Scenario 1 : 80% sensitive (b) Scenario 2 : 60% sensitive (c) Scenario 3 : 20% sensitive

Figure 9: Fraction of successful MTC devices in heterogeneous scenarios for different delay budgets. $N = 30000$ and $N_{pr} = 40$.



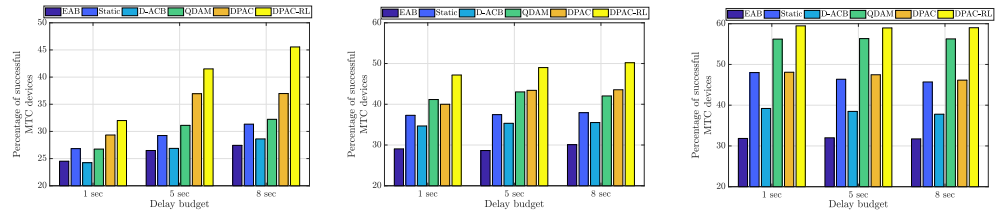
(a) Scenario 1 : 80% sensitive (b) Scenario 2 : 60% sensitive (c) Scenario 3 : 20% sensitive

Figure 10: Fraction of successful MTC devices in heterogeneous scenarios for different delay budgets. $N = 60000$ and $N_{pr} = 40$.



(a) Scenario 1 : 80% sensitive (b) Scenario 2 : 60% sensitive (c) Scenario 3 : 20% sensitive

Figure 11: Fraction of successful MTC devices in heterogeneous scenarios for different delay budgets. $N = 90000$ and $N_{pr} = 40$.



(a) Scenario 1 : 80% sensitive (b) Scenario 2 : 60% sensitive (c) Scenario 3 : 20% sensitive

Figure 12: Fraction of successful MTC devices in heterogeneous scenarios for different delay budgets. $N = 30000$ and $N_{pr} = 10$.

the percentage of delay-sensitive devices are: 80% (Scenario 1), 60% (Scenario 2), 20% (Scenario 3). In the heterogeneous scenario, the performance of the proposed DPAC based ACB and DPAC-RL framework is compared with EAB [13], static ACB, D-ACB [19], as well as with QDAM [27] that specifically considered heterogeneous traffic.

The percentage of successful MTC devices in three different scenarios is studied in Figs. 9, 10, and 11 for number of preambles $N_{pr} = 40$ and total number of arrivals $N = 30000, 60000$, and 90000, respectively. Figs. 9 and 12 gives a comparative picture of performance number of preambles N_{pr} is varied from 40 to 10, while total number of arrivals is kept constant at $N = 30000$. It can be seen that, DPAC has better throughput performance compared to EAB, static ACB and D-ACB. Whereas, QDAM performs as good as or even better than DPAC in some cases. But it is evident from Figs. 9 - 12 that, DPAC-RL beats all other comparative schemes

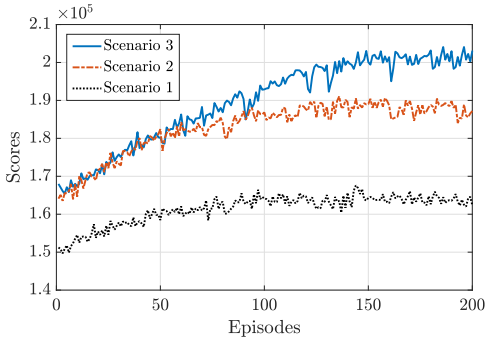


Figure 13: Convergence of DPAC-RL with $N = 60000$, $N_{pr} = 40$, $d_B = 1$ s for different heterogeneous scenarios.

in all scenarios, with performance improvement being up to $\sim 60\%$ compared to both D-ACB and QDAM in terms of percentage of MTC devices successful in preamble transmission. Intuitively, the reason behind the success of DPAC-RL is its ability to continuously learn by taking feedback from the dynamic environment.

Time-complexity of the proposed DPAC-RL scheme depends on the complexity of DQN which is given by the number of episodes taken by the agent to converge [39]. In Fig. 13, the number of episodes taken by DPAC-RL to converge in different scenarios with $N = 60000$ and $N_{pr} = 40$ is shown . It can be seen that in all scenarios, DQN converges after 200 episodes. The complexity of the DPAC scheme is primarily determined by the optimization framework described in Section IV. The optimization problem in (17) was solved in MATLAB using interior point methods which has time complexity of $O(\mathfrak{N}_d^3\mathfrak{L})$, where $\mathfrak{N}_d$ is dimension of the vector being optimized and $\mathfrak{L}$ is the bit length of input data [40]. In DPAC, dimension of the vector being optimized is 3, and the default bit length used in MATLAB is 32. Consequently, the time complexity of the optimization in Section IV is $O(1)$.

VII. Conclusion

In this paper a novel DPAC-based ACB scheme has been proposed towards access congestion control in LTE random access for delay-constrained mMTC applications. The proposed framework assigns higher priority and hence higher success probability to those MTC devices that are close to crossing their respective delay budgets. An optimization framework has been used to find optimal model parameters that maximize the system utility function. Using Lagrangian duality, a lower bound for the utility has been obtained. Extensive simulations have been performed to evaluate efficiency of the proposed DPAC in homogeneous as well as heterogeneous QoS scenarios. Compared to state-of-the-art D-ACB algorithm and the conventional static ACB approach, performance of the proposed DPAC scheme in terms of successful preamble transmission has been found to gain up to $\sim 40\%$ and $\sim 50\%$, respectively, in homogeneous traffic scenario. In heterogeneous scenario, where the percentage of delay-sensitive MTC devices is varied, the gain is up to $\sim 30\%$. Further, the proposed DQN based online learning framework (DPAC-RL) is able to achieve up to $\sim 75\%$ improvement over D-ACB with homogeneous arrival and up to $\sim 60\%$ gain over closest competitive QDAM algorithm in heterogeneous scenario.

Appendix A
Derivation of CDF of delay due to ACB

As mentioned in Section III-B, D is the RV that defines the delay due to the ACB scheme in sub-frames. Let $\mathcal{T}$ be the RV defining the number of RAOs that the first preamble transmission of an MTC device is delayed due to ACB, i.e., the delay induced by ACB in terms of number of RAOs. Thus, the PDF of delay D can be obtained as:

$$f_D(it_{\text{rao}}) = \Pr\{D = it_{\text{rao}}\} = \Pr\{\mathcal{T} = i\} = f_{\mathcal{T}}(i). \quad \text{(A.1)}$$

Let Y be the RV in the domain y that represents the number of barring checks performed by an MTC device. If the MTC device succeeds in its very first barring check, i.e., $(Y = 1)$, the preamble is transmitted immediately. Probability of this event is p_{acb}, and the PMF of $\mathcal{T}$ given $Y = 1$ is: $f_{\mathcal{T}|Y}(i \mid 1) = \delta(i)$, where $\delta(\cdot)$ is the Dirac delta function.

Also, the PMF of $\mathcal{T}$ given $Y = 2$ is positive between $i_{\mathcal{T},\min} = \left\lceil \frac{0.7t_{\text{acb}}}{t_{\text{rao}}} \right\rceil$ and $i_{\mathcal{T},\max} = \left\lceil \frac{1.3t_{\text{acb}}}{t_{\text{rao}}} \right\rceil$. Its PMF is:

$$f_{\mathcal{T}|Y}(i \mid 2) = \frac{1}{0.6t_{\text{acb}}} \begin{cases} it_{\text{rao}} - 0.7t_{\text{acb}}, i = i_{\mathcal{T},\min} \\ t_{\text{rao}}, i_{\mathcal{T},\min} < i < i_{\mathcal{T},\max} \\ 1.3t_{\text{acb}} - (i-1)t_{\text{rao}}, i = i_{\mathcal{T},\max}. \end{cases} \quad \text{(A.2)}$$

PMF of $\mathcal{T}$ given $Y > 2$ can be calculated recursively as:

$$f_{\mathcal{T}|Y}(i \mid y > 2) = \sum_{\ell = i_{\mathcal{T},\min}}^{i_{\mathcal{T},\max}} f_{\mathcal{T}|Y}\left(\ell \mid 2\right) f_{\mathcal{T}|Y}(i - \ell \mid y - 1). \quad \text{(A.3)}$$

The probability of an MTC device succeeding in Y barring checks is given by: $f_Y(y) = p_{\text{acb}}\left(1 - p_{\text{acb}}\right)^{y-1}, \quad \forall y = 1, 2, \cdots$, from which PMF of $\mathcal{T}$ can then be calculated as:

$$f_{\mathcal{T}}(i) = \sum_{y=1}^{\infty} f_{\mathcal{T}|Y}(i \mid y) f_Y(y), \qquad \text{for } i = 0, 1, 2, \ldots \quad \text{(A.4)}$$

To truncate the infinite sample spaces of Y, the maximum number of allowed barring checks $y_{\max}$ is found [17]. Hence the MTC devices, which fail in the first $y_{\max}$ barring checks, put an end to ACB and do not take part in the random access contention. $y_{\max}$ is calculated as: $y_{\max} = \left\lceil \frac{\log p_{\mathcal{E}_{\text{acb}}}}{\log(1 - p_{\text{acb}})} \right\rceil$ where $p_{\mathcal{E}_{\text{acb}}}$, the probability that an MTC device terminates the ACB scheme, is given as: $p_{\mathcal{E}_{\text{acb}}} = \left(1 - p_{\text{acb}}\right)^{y_{\max}}$. A certain value for $p_{\mathcal{E}_{\text{acb}}}$ is chosen empirically, which truncates the summation up to some y_{max}. By truncating (A.4), $f_{\mathcal{T}}(i)$ is approximated as:

$$\begin{aligned} f_{\mathcal{T}'}(i) &= f_{\mathcal{T}|Y \le y_{\max}}(i) \\ &= \frac{1}{1 - \left(1 - p_{\text{acb}}\right)^{y_{\max}}} \sum_{y=1}^{y_{\max}} f_{\mathcal{T}|Y}(i \mid y) f_Y(y). \end{aligned}$$

It is notable that $f_{\mathcal{T}'}(i)$ is a PDF, and in fact, $f_{\mathcal{T}}(i) \approx f_{\mathcal{T}'}(i)$, if $p_{\mathcal{E}_{\text{acb}}} \ll 1$. Hence, the PDF of delay due to ACB $f_D(\cdot)$ is computed using the approximated PMF $f_{\mathcal{T}'}(\cdot)$ and then the CDF is obtained as: $F_D(d) = \int_{-\infty}^{d} f_D(d)$.

Appendix B
Recursion to derive arrival distribution

Let S and C be the RVs denoting the number of preambles transmitted respectively by one successful and by multiple unsuccessful MTC devices. Their joint probability distribution for a given $n(i), p_{\text{S,C}}\left(s, c; n(i)\right)$ is recursively calculated as:

$$\begin{aligned} &p_{S,C}\left(s, c; n(i)\right) \\ &= \left(\frac{N_{pr} - s + 1 - c}{N_{pr}}\right) p_{S,C}\left(s - 1, c; n(i) - 1\right) \quad \text{(B.1)} \\ &+ \frac{c}{N_{pr}} p_{S,C}\left(s, c; n(i) - 1\right) + \frac{s+1}{N_{pr}} p_{S,C}\left(s+1, c-1; n(i) - 1\right) \end{aligned}$$

for $s = 0, 1, \cdots, s_{\max}$, and $c = 0, 1, \cdots, c_{\max}$ with the initial condition $p_{S,C}(0, 0; 0) = 1$. The marginal PMFs of S and C

for a given $n(i)$, i.e., $p_S(s; n(i))$ and $p_C(c; n(i))$ are calculated by integrating their joint distribution.

Let $R_{\mathrm{S}}(i)$ and $R_{\mathrm{C}}(i)$ be the RVs that define the number of preambles transmitted by exactly one and by multiple MTC devices at the i^{th} RAO, respectively. Their respective PMFs are derived from the PMFs of S and C by linear interpolation.

Decoding probability for the k^{th} transmitted preamble by an MTC device is $p_{\mathcal{D};k}$, modeled in [9] as:

$$p_{D;k} = 1 - \frac{1}{e^k}. \tag{B.2}$$

The average preamble detection probability at the i^{th} RAO, by a little abuse of notation from (B.2), is denoted as:

$$p_{D:i} = \frac{1}{\mathbb{E}\left[n(i)\right]} \sum_{k=1}^{k_{\max}} p_{D;k} \mathbb{E}\left[n(i,k)\right]. \tag{B.3}$$

Subsequently, the expected value of number of decoded preambles at the i^{th} RAO, $R_D(i)$, and the number of MTC devices that will receive an uplink grant in response to a preamble transmitted at the i^{th} RAO, $M_U(i)$, are obtained using their respective PMFs defined in [17].

Intuitively, $\mathbb{E}\left[M_{\mathrm{U}}(i)\right]$ is indeed the expected number of MTC devices to successfully complete the first two steps of the random access procedure. Hence, the expected number of devices successfully completing the first two steps of random access in their k^{th} preamble transmission is obtained as:

$$\mathbb{E}\left[M_{\mathrm{U}}(i,k)\right] = \frac{\mathbb{E}\left[M_{\mathrm{U}}(i)\right]\mathbb{E}\left[n(i,k)\right]p_{D;k}}{\mathbb{E}\left[n(i)\right]p_{D,i}}.$$

Then, the expected number of failed accesses is found as:

$$\mathbb{E}\left[M_{\mathrm{F}}(i,k)\right] = \mathbb{E}\left[n(i,k)\right] - \mathbb{E}\left[M_{\mathrm{U}}(i,k)\right].$$

The conditional PMF $p_{B|K}$, defined in [17], is used to model the back-off process at each RAO by means of the following recursion, where B is the RV denoting the number of RAOs that an MTC device has to wait due to back-off and K is the RV denoting the number of preamble transmissions by the device:

$$\mathbb{E}\left[n(i,k)\right] = \sum_{j=j_{\min}}^{j_{\max}} \mathbb{E}\left[M_{\mathrm{F}}(i-j, k-1)\right] p_{B|K}(j \mid 2) \tag{B.4}$$

where $i = 1, 2, \cdots, i_{\max}, k = 2, 3, \cdots, k_{\max}$, $i_{\max} = t_{\text{dist}} + (k_{\max}-1)\, i_{B,\max} + (x_{\max}-1)\, i_{T,\max}$ is the last RAO when a preamble transmission can occur, $j_{\min} = \min\{i_{B,\min}, i\}$, $j_{\max} = \min\{i_{B,\max}, i\}$, and $\mathbb{E}\left[n(1,k)\right] = 0$.

Appendix C
Proof of Lemma 1

Here, an upper bound of the expected number of successful preamble transmissions for a given number of contending MTC devices n is obtained. The expected number of successful preamble transmissions for a given n is given by:

$$E[\Upsilon_i | \Lambda_i = n] = n\bar{p}\left(1 - \frac{\bar{p}}{N_{pr}}\right)^{n-1}. \tag{C.1}$$

To maximize (C.1), the objective function $f_{0,T} = -n\bar{p}\left(1-\frac{\bar{p}}{N_{pr}}\right)^{n-1}$ is minimized over the feasible set defined by the constraints in (17). To find the lower bound, Lagrangian duality is used. The Lagrangian function $\mathcal{L}$ is expressed as:

$$\begin{aligned}\mathcal{L}(\bar{x}, \bar{\lambda}) =& f_{0,T} + \sum_{k=1}^{K} \left(\lambda_k^+(p_k - 1) + \lambda_k^-(-p_k)\right) \\ &+ \lambda_{K+1}^+(\bar{p} - 1) + \lambda_{K+1}^-(\frac{N_{pr}}{n} - \bar{p})\end{aligned}$$

where the Lagrange multipliers are the components of the vector $\bar{\lambda} = [\lambda_1^+, \cdots, \lambda_K^+, \lambda_1^-, \cdots, \lambda_K^-, \lambda_{K+1}^+, \lambda_{K+1}^-]$.

The derivatives of $\bar{p} = \sum_{k=1}^{K} c_k(x_1 k^{x_2} + x_3)$ with respect to the optimization variables can be computed as:

$$\frac{\partial \bar{p}}{\partial x_1} = \sum_{k=1}^{K} c_k k^{x_2}, \quad \frac{\partial \bar{p}}{\partial x_2} = \sum_{k=1}^{K} c_k x_1 k^{x_2} \ln k, \quad \frac{\partial \bar{p}}{\partial x_3} = \sum_{k=1}^{K} c_k.$$

$$\text{Also, } \frac{\partial f_{0,T}}{\partial \bar{p}} = n\left(1 - \frac{\bar{p}}{N_{pr}}\right)^{n-2}\left(\frac{n\bar{p}}{N_{pr}} - 1\right). \tag{C.2}$$

Using the chain rule, the gradient of $\mathcal{L}$ can be computed as:

$$\nabla\mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x_1} \\ \frac{\partial \mathcal{L}}{\partial x_2} \\ \frac{\partial \mathcal{L}}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{K}\left(c_k\xi + \lambda_k^+ - \lambda_k^-\right)k^{x_2} \\ \sum_{k=1}^{K}\left(c_k\xi + \lambda_k^+ - \lambda_k^-\right)x_1 k^{x_2}\ln k \\ \sum_{k=1}^{K}\left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) \end{bmatrix}$$

where $\xi = \left(\frac{\partial f_{0,T}}{\partial \bar{p}} + \lambda_{K+1}^+ - \lambda_{K+1}^-\right)$.

To satisfy the Karush–Kuhn–Tucker (KKT) conditions,

$$\begin{aligned}&\nabla\mathcal{L} = 0 \implies \sum_{k=1}^{K}\left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) = 0 \implies (\frac{\partial f_{0,T}}{\partial \bar{p}})c_k^{sum} + \\ &\qquad \sum_{k=1}^{K}\left(\lambda_k^+ - \lambda_k^-\right) + \left(\lambda_{K+1}^+ - \lambda_{K+1}^-\right)c_k^{sum} = 0 \\ &\implies (\frac{\partial f_{0,T}}{\partial \bar{p}})c_k^{sum} + \bar{w}\bar{\lambda} = 0, \text{ where } c_k^{sum} = \sum_{k=1}^{K} c_k \text{ and} \\ &\qquad \bar{w} = [1, \cdots, 1, -1, \cdots, -1, c_k^{sum}, -c_k^{sum}]. \\ &\implies n\left(1 - \frac{\bar{p}}{N_{pr}}\right)^{n-2}\left(\frac{n\bar{p}}{N_{pr}} - 1\right) + \frac{\bar{w}\bar{\lambda}}{c_k^{sum}} = 0 \\ &\implies \left(1 - \frac{\bar{p}(n-2)}{N_{pr}}\right)\left(1 - \frac{n\bar{p}}{N_{pr}}\right) = \frac{\bar{w}\bar{\lambda}}{nc_k^{sum}} \\ &\implies \left(\bar{p} - \frac{N_{pr}}{n-2}\right)\left(\bar{p} - \frac{N_{pr}}{n}\right) = \frac{\bar{w}\bar{\lambda}}{nc_k^{sum}}.\end{aligned}$$

The roots of $(\bar{p} - \alpha)(\bar{p} - \beta) = \gamma$ are:

$$\bar{p} = \frac{\alpha+\beta}{2} \pm \frac{1}{2}\sqrt{(\alpha-\beta)^2 + 4\gamma} = \frac{\alpha+\beta}{2} \pm \sqrt{\frac{(\alpha-\beta)^2}{4} + \gamma}$$

$$\text{where } \alpha = \frac{N_{pr}}{n-2}, \beta = \frac{N_{pr}}{n}, \gamma = \frac{\bar{w}\bar{\lambda}}{nc_k^{sum}}.$$

$$\therefore \alpha + \beta = \frac{N_{pr}}{n-2} + \frac{N_{pr}}{n} = \frac{2N_{pr}(n-1)}{n(n-2)}; \alpha - \beta = \frac{2N_{pr}}{n(n-2)}$$

$$\text{Hence, } \bar{p}(\bar{x}^*) = \frac{N_{pr}(n-1)}{n(n-2)} \pm \sqrt{\frac{{N_{pr}}^2}{n^2(n-2)^2} + \frac{\bar{w}\bar{\lambda}}{nc_k^{sum}}}.$$

By complementary slackness condition,

$$\lambda_k^{+*}(p_k(\bar{x}^*) - 1) = 0, \lambda_k^{-*}(-p_k(\bar{x}^*)) = 0,$$

$$\lambda_{K+1}^{+^*}(\bar{p}(\bar{x}^*)-1)=0, \lambda_{K+1}^{-^*}\left(\frac{N_{pr}}{n}-\bar{p}(\bar{x}^*)\right)=0. \quad \text{(C.3)}$$

The dual function can be written as:

$$g_d(\bar{\lambda})=\min_{\bar{x}}\mathcal{L}(\bar{x},\bar{\lambda})=\mathcal{L}(\bar{x}^*,\bar{\lambda})=-n\bar{p}(\bar{x}^*)\left(1-\frac{\bar{p}(\bar{x}^*)}{N_{pr}}\right)^{n-1}.$$

To maximize $g_d(\bar{\lambda})$, the dual problem is formulated as,

$$(D): \max_{\bar{\lambda}} g_d(\bar{\lambda})=\mathcal{L}(\bar{x}^*,\bar{\lambda}) \quad \text{s.t. } \bar{\lambda}\geq 0. \quad \text{(C.4)}$$

The maxima of the dual objective correspond to the value of $\bar{\lambda}$, for which:

$$\frac{\partial g_d(\bar{\lambda})}{\partial\bar{\lambda}}=0, \text{ i.e., } \frac{\partial}{\partial\bar{\lambda}}\mathcal{L}(\bar{x}^*,\bar{\lambda})=0$$
$$\implies \left(n\left(1-\frac{\bar{p}(\bar{x}^*)}{N_{pr}}\right)^{n-2}\left(\frac{n\bar{p}(\bar{x}^*)}{N_{pr}}-1\right)\right)\frac{\partial\bar{p}(\bar{x}^*)}{\partial\bar{\lambda}}=0$$
$$\implies \bar{p}(\bar{x}^*)=\frac{N_{pr}}{n}$$
$$\implies \frac{N_{pr}(n-1)}{n(n-2)}\pm\sqrt{\frac{{N_{pr}}^2}{n^2(n-2)^2}+\frac{\bar{w}\bar{\lambda}}{nc_k^{sum}}}=\frac{N_{pr}}{n}$$
$$\implies \pm\sqrt{\frac{1}{(n-2)^2}+\frac{n^2\bar{w}\bar{\lambda}}{nc_k^{sum}{N_{pr}}^2}}=-\frac{1}{n-2}$$
$$\implies \bar{\lambda}^*=0 \implies \bar{p}(\bar{x}^*)\Big|_{\bar{\lambda}=\bar{\lambda}^*}=\frac{N_{pr}(n-1)}{n(n-2)}\pm\frac{N_{pr}}{n(n-2)}$$
$$\implies \bar{p}(\bar{x}^*)\Big|_{\bar{\lambda}=\bar{\lambda}^*}=\frac{N_{pr}}{n-2}, \text{ (OR), } \bar{p}(\bar{x}^*)\Big|_{\bar{\lambda}=\bar{\lambda}^*}=\frac{N_{pr}}{n}.$$

$$\text{Hence, } d^*=\max_{\bar{\lambda}} g_d(\bar{\lambda})=\max_{\bar{\lambda}}\mathcal{L}(\bar{x}^*,\bar{\lambda})=\mathcal{L}(\bar{x}^*,\bar{\lambda}^*)$$
$$=\left(-n\bar{p}(\bar{x}^*)\left(1-\frac{\bar{p}(\bar{x}^*)}{N_{pr}}\right)^{n-1}\right)\Big|_{\bar{\lambda}=\bar{\lambda}^*}$$
$$=-\frac{nN_{pr}}{n-2}\left(1-\frac{1}{n-2}\right)^{n-1}, \forall \quad n>2. \quad \text{(C.5)}$$

Thus, $E[\Upsilon_i|\Lambda_i = n]$ is upper bounded by $\frac{nN_{pr}}{n-2}\left(1-\frac{1}{n-2}\right)^{n-1}$ for a given number of arrivals n (when $n\leq 2$, anyway the DPAC algorithm will not be enabled as $p_k=1$ is assigned in those instances where $n\leq N_{pr}$) and number of available preambles N_{pr}.

APPENDIX D
PROOF OF LEMMA 2

It is shown here that the primal problem (P) in (17) is quasi-convex in the model parameter vector $\bar{x}=[x_1,x_2,x_3]$.

1) Set of constraints C_1: In the set of inequality constraints C_1, the constraint functions are given by:

$$p_k(\bar{x})=p_k(x_1,x_2,x_3)=(x_1k^{x_2}+x_3), \quad \forall k\in\{1,2,\cdots K\}.$$

First bordered Hessian matrix of p_k is:

$$\mathbf{H}_{\mathbf{p_k}}^{\mathbf{B}\ (1)}=\begin{bmatrix}0 & p'_{k1}(\bar{x})\\ p'_{k1}(\bar{x}) & p''_{k11}(\bar{x})\end{bmatrix}=\begin{bmatrix}0 & k^{x_2}\\ k^{x_2} & 0\end{bmatrix}.$$
$\therefore D_{1,p_k} =$ Determinant of first bordered Hessian matrix
$$=-(k^{x_2})^2<0.$$

Second bordered Hessian matrix of p_k is:

$$\mathbf{H}_{\mathbf{p_k}}^{\mathbf{B}\ (2)}=\begin{bmatrix}0 & p'_{k1}(\bar{x}) & p'_{k2}(\bar{x})\\ p'_{k1}(\bar{x}) & p''_{k11}(\bar{x}) & p''_{k12}(\bar{x})\\ p'_{k2}(\bar{x}) & p''_{k21}(\bar{x}) & p''_{k22}(\bar{x})\end{bmatrix}=\begin{bmatrix}0 & k^{x_2} & x_1k^{x_2}\ln k\\ k^{x_2} & 0 & k^{x_2}\ln k\\ x_1k^{x_2}\ln k & k^{x_2}\ln k & x_1k^{x_2}(\ln k)^2\end{bmatrix}.$$
$\therefore D_{2,p_k} =$ Determinant of second bordered Hessian matrix
$$=x_1(k^{x_2})^3(\ln k)^2\geqslant 0, \forall x_1\geqslant 0.$$

Third bordered Hessian matrix of p_k:

$$\mathbf{H}_{\mathbf{p_k}}^{\mathbf{B}\ (3)}=\begin{bmatrix}0 & p'_{k1}(\bar{x}) & p'_{k2}(\bar{x}) & p'_{k3}(\bar{x})\\ p'_{k1}(\bar{x}) & p''_{k11}(\bar{x}) & p''_{k12}(\bar{x}) & p''_{k13}(\bar{x})\\ p'_{k2}(\bar{x}) & p''_{k21}(\bar{x}) & p''_{k22}(\bar{x}) & p''_{k23}(\bar{x})\\ p'_{k3}(\bar{x}) & p''_{k31}(\bar{x}) & p''_{k32}(\bar{x}) & p''_{k33}(\bar{x})\end{bmatrix}=\begin{bmatrix}0 & k^{x_2} & x_1k^{x_2}\ln k & 1\\ k^{x_2} & 0 & k^{x_2}\ln k & 0\\ x_1k^{x_2}\ln k & k^{x_2}\ln k & x_1k^{x_2}(\ln k)^2 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}.$$
$\therefore D_{3,p_k} =$ Determinant of third bordered Hessian matrix $=0$.

Therefore, $D_{n,p_k}\leqslant 0$ when n is ODD and $D_{n,p_k}\geqslant 0$ when n is EVEN; hence, p_k is quasi-concave in x_1,x_2,x_3. This can also be proven from the concept of level set. The super level set of p_k is convex. Hence, p_k is quasi-concave when $x_1\geqslant 0$.

2) Constraint C_2: In the inequality constraint C_2, the constraint function $\bar{p}$ is given by:

$$\bar{p}(\bar{x})=\bar{p}(x_1,x_2,x_3)=\sum_{k=1}^{K}c_kp_k=\sum_{k=1}^{K}c_k(x_1k^{x_2}+x_3).$$

First bordered Hessian matrix of $\bar{p}$ is:

$$\mathbf{H}_{\bar{\mathbf{p}}}^{\mathbf{B}(1)}=\begin{bmatrix}0 & \bar{p}'_1(\bar{x})\\ \bar{p}'_1(\bar{x}) & \bar{p}''_{11}(\bar{x})\end{bmatrix}=\begin{bmatrix}0 & \sum_{k=1}^{K}c_kk^{x_2}\\ \sum_{k=1}^{K}c_kk^{x_2} & 0\end{bmatrix}.$$
$\therefore D_{1,\bar{p}} =$ Determinant of first bordered Hessian matrix
$$=-\left(\sum_{k=1}^{K}c_kk^{x_2}\right)^2<0.$$

Second bordered Hessian matrix of $\bar{p}$ is:

$$\mathbf{H}_{\bar{\mathbf{p}}}^{\mathbf{B}(2)}=\begin{bmatrix}0 & \bar{p}'_1(\bar{x}) & \bar{p}'_2(\bar{x})\\ \bar{p}'_1(\bar{x}) & \bar{p}''_{11}(\bar{x}) & \bar{p}''_{12}(\bar{x})\\ \bar{p}'_2(\bar{x}) & \bar{p}''_{21}(\bar{x}) & \bar{p}''_{22}(\bar{x})\end{bmatrix}$$

where $\bar{p}'_2(\bar{x})=x_1\sum_{k=1}^{K}c_kk^{x_2}\ln k,$

$$\bar{p}''_{12}(\bar{x})=\bar{p}''_{21}(\bar{x})=\sum_{k=1}^{K}c_kk^{x_2}\ln k, \text{ and}$$
$$\bar{p}''_{22}(\bar{x})=x_1\sum_{k=1}^{K}c_kk^{x_2}(\ln k)^2$$

$\therefore D_{2,\bar{p}} =$ Determinant of second bordered Hessian matrix
$$=x_1\left(\sum_{k=1}^{K}c_kk^{x_2}\ln k\right)\left(\sum_{k=1}^{K}c_kk^{x_2}\right)\left(\sum_{k=1}^{K}c_kk^{x_2}\ln k\right)$$

$\geqslant 0, \forall x_1 \geqslant 0.$

Similar to the case of p_k,
$D_{3,\bar{p}} =$ Determinant of third bordered Hessian matrix $= 0$. Therefore, $\bar{p}$ is also quasi-concave when $x_1 \geqslant 0$.

3) Objective function f_0: If g is quasiconcave and h is non-increasing real-valued function on the real line, then $f = h \circ g$ is quasiconvex. The non-increasing nature of f_0 is evident from Fig. 4 by observation. Hence, $\bar{p}$ being a quasiconcave function and f_0 a non-increasing real-valued function, f_0 is quasiconvex in $\{x_1, x_2, x_3\}$. Hence the primal problem (P) is a quasiconvex optimization problem which can be solved by a family of convex feasibility problems [41].

Appendix E
Proof of Lemma 3

Here a lower bound of the primal objective function f_0 is obtained. The primal problem (P) is transformed into its dual problem (D) to accommodate the constraints. Then Lagrangian duality [41] is used to obtain a lower bound, i.e., the minimum value, the optimal solution of primal problem p^* can achieve.

The Lagrangian function $\mathcal{L}$ can be expressed as:

$$\mathcal{L}(\bar{x}, \bar{\lambda}) = f_0 + \sum_{k=1}^{K} \left(\lambda_k^+(p_k - 1) + \lambda_k^-(-p_k)\right) + \lambda_{K+1}^+(\bar{p} - 1) +$$

$$\lambda_{K+1}^-(\frac{N_{pr}}{n} - \bar{p}) = \alpha\bar{p}^3 + \beta\bar{p}^2 + \gamma\bar{p} + \delta + \sum_{k=1}^{K}(\lambda_k^+(p_k - 1) +$$

$$\lambda_k^-(-p_k)) + \lambda_{K+1}^+(\bar{p} - 1) + \lambda_{K+1}^-(\frac{N_{pr}}{n} - \bar{p}).$$

The derivatives of $\bar{p} = \sum_{k=1}^{K} c_k(x_1 k^{x_2} + x_3)$ with respect to the optimization variables can be computed as:

$$\frac{\partial \bar{p}}{\partial x_1} = \sum_{k=1}^{K} c_k k^{x_2}, \quad \frac{\partial \bar{p}}{\partial x_2} = \sum_{k=1}^{K} c_k x_1 k^{x_2} \ln k, \quad \frac{\partial \bar{p}}{\partial x_3} = \sum_{k=1}^{K} c_k.$$

Using the chain rule, the gradient of $\mathcal{L}$ can be computed as:

$$\nabla\mathcal{L} = \begin{bmatrix} \sum_{k=1}^{K} \left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) k^{x_2} \\ \sum_{k=1}^{K} \left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) x_1 k^{x_2} \ln k \\ \sum_{k=1}^{K} \left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) \end{bmatrix},$$

where $\xi = \left(3\alpha\bar{p}^2 + 2\beta\bar{p} + \gamma + \lambda_{K+1}^+ - \lambda_{K+1}^-\right)$.

For the KKT conditions to be satisfied,

$$\nabla\mathcal{L} = 0 \implies \sum_{k=1}^{K} \left(c_k\xi + \lambda_k^+ - \lambda_k^-\right) = 0$$

$$\implies (3\alpha\bar{p}^2 + 2\beta\bar{p} + \gamma)c_k^{sum} + \sum_{k=1}^{K} \left(\lambda_k^+ - \lambda_k^-\right)$$

$$+ \left(\lambda_{K+1}^+ - \lambda_{K+1}^-\right) c_k^{sum} = 0$$

$$\implies (3\alpha\bar{p}^2 + 2\beta\bar{p} + \gamma)c_k^{sum} + \bar{w}\bar{\lambda} = 0$$

where $c_k^{sum} = \sum_{k=1}^{K} c_k$, and $\bar{w} = [1, \cdots, 1, -1, \cdots, -1, c_k^{sum}, -c_k^{sum}]$

$$\implies 3\alpha\bar{p}^2 + 2\beta\bar{p} + \left(\gamma + \frac{\bar{w}\bar{\lambda}}{c_k^{sum}}\right) = 0$$

$$\therefore \bar{p}(\bar{x}^*) = \frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\left(\gamma + \frac{\bar{w}\bar{\lambda}}{c_k^{sum}}\right)}}{3\alpha}. \tag{E.1}$$

By complementary slackness condition,

$$\lambda_k^{+^*}(p_k(\bar{x}^*) - 1) = 0, \lambda_k^{-^*}(-p_k(\bar{x}^*)) = 0,$$

$$\lambda_{K+1}^{+^*}(\bar{p}(\bar{x}^*) - 1) = 0, \lambda_{K+1}^{-^*}\left(\frac{N_{pr}}{n} - \bar{p}(\bar{x}^*)\right) = 0. \tag{E.2}$$

The dual function can be written as:

$$g_d(\bar{\lambda}) = \min_{\bar{x}} \mathcal{L}(\bar{x}, \bar{\lambda}) = \mathcal{L}(\bar{x}^*, \bar{\lambda})$$

$$= \alpha\bar{p}^3(\bar{x}^*) + \beta\bar{p}(\bar{x}^*)^2 + \gamma\bar{p}(\bar{x}^*) + \delta. \tag{E.3}$$

To maximize $g_d(\bar{\lambda})$, the dual problem is formulated as:

$$(D) : \max_{\bar{\lambda}} g_d(\bar{\lambda}) = \mathcal{L}(\bar{x}^*, \bar{\lambda}) \quad \text{s.t. } \bar{\lambda} \geq 0. \tag{E.4}$$

The maxima of $g_d(\bar{\lambda})$ corresponds to value of $\bar{\lambda}$, for which:

$$\frac{\partial g_d(\bar{\lambda})}{\partial\bar{\lambda}} = 0, \text{ i.e., } \frac{\partial}{\partial\bar{\lambda}}\mathcal{L}(\bar{x}^*, \bar{\lambda}) = 0$$

$$\implies \left(3\alpha\bar{p}^2(\bar{x}^*) + 2\beta\bar{p}(\bar{x}^*) + \gamma\right)\frac{\partial\bar{p}(\bar{x}^*)}{\partial\bar{\lambda}} = 0$$

$$\implies \bar{p}(\bar{x}^*)\Big|_{\bar{\lambda}=\bar{\lambda}^*} = \frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}$$

$$\therefore d^* = \max_{\bar{\lambda}} g_d(\bar{\lambda}) = \max_{\bar{\lambda}} \mathcal{L}(\bar{x}^*, \bar{\lambda}) = \mathcal{L}(\bar{x}^*, \bar{\lambda}^*)$$

$$= \left(\alpha\bar{p}^3(\bar{x}^*) + \beta\bar{p}(\bar{x}^*)^2 + \gamma\bar{p}(\bar{x}^*) + \delta\right)\Big|_{\bar{\lambda}=\bar{\lambda}^*}$$

$$= \alpha\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right)^3 + \beta\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right)^2$$

$$+ \gamma\left(\frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha}\right) + \delta. \tag{E.5}$$

If p^* is the solution to the primal problem (P), then $d^* \leqslant p^*$, i.e., d^* is the lower bound of the primal minimization problem.

References

[1] F. Ghavimi and H. Chen, "M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 525–549, Second quarter 2015.

[2] "Service Requirements for Machine-Type Communications (MTC)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.368, Dec. 2016, version 13.2.0.

[3] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive Multivariate Data Compression in Smart Metering Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1287–1297, 2021.

[4] S. Tripathi and S. De, "Channel-Adaptive Transmission Protocols for Smart Grid IoT Communication," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7823–7835, 2020.

[5] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[6] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 4–16, First quarter 2014.

[7] "Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.331, Aug. 2016, version 13.2.0.

[8] S. Tripathi and S. De, "Dynamic Prediction of Powerline Frequency for Wide Area Monitoring and Control," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2837–2846, 2018.

[9] "Study on RAN Improvements for Machine-Type Communications," 3rd Generation Partnership Project (3GPP), Sophia Antipolis Cedex, France, Technical Report (TR) 37.868, Aug. 2011, version 11.0.0.

[10] L. Tello-Oquendo, I. Leyva-Mayorga, V. Pla, J. Martinez-Bauset, J. 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 Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3505–3520, Apr. 2018.

[11] G. Lin, S. Chang, and H. Wei, "Estimation and Adaptation for Bursty LTE Random Access," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2560–2577, Apr. 2016.

[12] C. Wei, G. Bianchi, and R. Cheng, "Modeling and analysis of random access channels with bursty arrivals in OFDMA wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 1940–1953, 2015.

[13] R. Cheng, J. Chen, D. Chen, and C. Wei, "Modeling and analysis of an extended access barring algorithm for machine-type communications in LTE-A networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 2956–2968, Jun. 2015.

[14] O. Arouk and A. Ksentini, "General model for RACH procedure performance analysis," *IEEE Commun. Lett.*", vol. 20, no. 2, pp. 372–375, Feb. 2016.

[15] M. Koseoglu, "Lower bounds on the LTE-A average random access delay under massive M2M arrivals," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2104–2115, May 2016.

[16] O. Galinina, A. Turlikov, T. Tirronen, J. Torsner, S. Andreev, and Y. Koucheryavy, "Random-access latency optimization and stability of highly-populated LTE-based M2M deployments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[17] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "On the accurate performance evaluation of the LTE-A random access procedure and the access class barring scheme," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 7785–7799, 2017.

[18] Z. Wang and V. W. S. Wong, "Optimal Access Class Barring for Stationary Machine Type Communication Devices With Timing Advance Information," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5374–5387, Oct. 2015.

[19] 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, Dec. 2016.

[20] H. Jin, W. T. Toor, B. C. Jung, and J. Seo, "Recursive pseudo-Bayesian access class Barring for M2M communications in LTE systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8595–8599, Sep. 2017.

[21] T. Lin, C. Lee, J. Cheng, and W. Chen, "PRADA: Prioritized random access with dynamic access barring for MTC in 3GPP LTE-A networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2467–2472, Jun. 2014.

[22] W. T. Toor and H. Jin, "Comparative study of access class barring and extended access barring for machine type communications," in *Proc. IEEE ICTC*, Oct. 2017, pp. 604–609.

[23] C. Di, B. Zhang, Q. Liang, S. Li, and Y. Guo, "Learning automata-based access class barring scheme for massive random access in machine-to-machine communications," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6007–6017, 2019.

[24] C.-H. Lee, S.-J. Kao, and F.-M. Chang, "LSTM-based ACB scheme for machine type communications in LTE-A networks," *Comput. Commun.*, vol. 152, pp. 296–304, 2020.

[25] D. Pacheco-Paramo and L. Tello-Oquendo, "Delay-aware dynamic access control for mMTC in wireless networks using deep reinforcement learning," *Comput. Netw.*, vol. 182, p. 107493, 2020.

[26] N. Jiang, Y. Deng, and A. Nallanathan, "Traffic prediction and random access control optimization: Learning and non-learning-based approaches," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 16–22, 2021.

[27] L. Zhao, X. Xu, K. Zhu, S. Han, and X. Tao, "QoS-based Dynamic Allocation and Adaptive ACB Mechanism for RAN Overload Avoidance in MTC," in *Proc. IEEE Global Commun. Conf*, Dec. 2018, pp. 1–6.

[28] P. Kansal and A. Bose, "Bandwidth and latency requirements for smart transmission grid applications," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1344–1352, Sep. 2012.

[29] S. De, C. Qiao, and S. Das, "A resource-efficient QoS routing protocol for mobile ad hoc networks," *Wirel. Commun. Mob. Comput.*, vol. 3, pp. 465–486, 2003.

[30] "Physical channels and modulation," 3GPP, TS 36.211,version 10.0.0 Release 10, Jan. 2011.

[31] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[32] M. Fernandez and S. Williams, "Closed-Form Expression for the Poisson-Binomial Probability Density Function," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, pp. 803–817, Apr. 2010.

[33] W. Ehm, "Binomial approximation to the Poisson binomial distribution," *Statistics & Probability Letters*, vol. 11, no. 1, pp. 7 – 16, 1991.

[34] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge: MIT press, 1998, vol. 1, no. 1.

[35] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," in *Proc. IEEE VTC*, Aug. 2018, pp. 1–6.

[36] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[37] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *KI 2010: Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer, 2010, pp. 203–210.

[38] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[39] J. Leem and H. Y. Kim, "Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning," *PloS one*, vol. 15(7):e0236178, Jul. 2020.

[40] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, no. 1, pp. 281–302, 2000, numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

Mayukh Roy Chowdhury received the B. Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology, Kolkata, India, in 2012 and the M.Tech degree in Communication Systems Engineering from Indian Institute of Technology Patna, India, in 2016. He is currently pursuing the Ph.D. degree with Department of Electrical Engineering, Indian Institute of Technology Delhi, India. His research interests include applied machine learning in 5G and next generation wireless networks, reinforcement learning driven radio resource management, AI on edge for smart IoT systems, random access for massive machine type communication in 5G, resource efficiency in communication networks.

Swades De (S'02 – M'04 – SM'14) received the B.Tech. degree in Radiophysics and Electronics from the University of Calcutta in 1993, the M.Tech. degree in Optoelectronics and Optical Communication from IIT Delhi in 1998, and the Ph.D. degree in Electrical Engineering from the State University of New York at Buffalo in 2004.

Dr. De is currently a Professor with the Department of Electrical Engineering, IIT Delhi. Before moving to IIT Delhi in 2007, he was a Tenure-Track Assistant Professor with the Department of ECE, New Jersey Institute of Technology, Newark, NJ, USA, from 2004–2007. He worked as an ERCIM Post-doctoral Researcher at ISTI-CNR, Pisa, Italy (2004), and has nearly five years of industry experience in India on telecom hardware and software development, from 1993–1997, 1999. His research interests are broadly in communication networks, with emphasis on performance modeling and analysis. Current directions include energy harvesting wireless networks, broadband wireless access and routing, network coexistence, smart grid networks, and IoT communications. Dr. De currently serves as an Area Editor of IEEE COMMUNICATIONS LETTERS and Elsevier Computer Communications, and an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and IEEE WIRELESS COMMUNICATIONS LETTERS.