A Comprehensive Distributed Queue-based Random Access Framework for mMTC in LTE/LTE-A Networks with Mixed-Type Traffic

Anh-Tuan H. Bui *Student Member, IEEE,* Chuyen T. Nguyen, Truong C. Thang *Member, IEEE,* and Anh T. Pham *Senior Member, IEEE*

***Abstract*—Thanks to their ubiquitous coverage, Long-Term Evolution (LTE) networks are considered the most potential enabler for massive Machine-Type Communications (mMTC) service in fifth-generation (5G) context. LTE standard, however, was not designed for mMTC and scenarios where the massive Machine-Type Devices (MTDs) population try to access a network over a short period may overload the Random Access CHannel (RACH). Furthermore, there is no mechanism to prioritize urgent MTDs in such overload situation. The baseline Access Class Barring (B-ACB) scheme is thus adopted by the 3GPP to address both issues at a substantial cost of access delay. This paper follows a different approach and proposes a complete solution to the two main issues of cellular mMTC. We promote the use of a mechanism called Distributed Queueing (DQ), aided by a MAC-layer load estimation technique, to effectively resolve contentions between the MTDs to improve delay performance with minimal impacts to LTE access procedure and air interface. Then, by exploiting information related to congestion level from the DQ process, a dynamic access prioritization scheme can be realized without additional signaling overhead. Computer simulation under an mMTC-oriented traffic model shows that our framework outperforms the B-ACB in terms of both access delay and energy consumption when all devices are of equal importance. On the other hand, when devices of different priorities coexist, our framework with proper tuning also offers lower delay for all classes and lower overall energy consumption compared to both the baseline and a dynamic ACB solutions in massive bursty access scenarios.**

***Index Terms*—Distributed Queue (DQ), Long-Term Evolution (LTE), massive Machine-Type Communications (mMTC), random access (RA)**

I. INTRODUCTION

The term massive Machine-Type Communications (mMTC) describes the prospect where a massive number of heterogenous Machine-Type Devices (MTDs) varying from ordinary surveillance cameras to unprecedented traffic sensors are connected to and exchange data/control information with application servers in an autonomous fashion i.e., requires little to no human intervention during deployment, operation and maintenance [1].

Expected to bring forth billions dollars worth of revenue to service providers over a wide range of revolutionary large-scale smart applications e.g., smart environment and intelligent transportation [1], mMTC research and development have recently gained more attention than ever. In fact, rapid advancements in wireless technologies have moved mMTC from a paper idea to a close reality, and supporting mMTC is now officially recognized as one of the main design objectives of the fifth-generation (5G) networks whose global rollout is due in 2020 [2]. Numerous standardization bodies have thus stepped in and leveraged their standards toward mMTC needs, most notably the *Institute of Electrical and Electronics Engineers* (IEEE)' enhancements to its broadband wireless access and personal area networks solution e.g., 802.16p, 802.15.4e, g and k [3], as well as the *3rd Generation Partnership Project* (3GPP)'s Narrowband-IoT (NB-IoT) cellular air interface optimized for low-rate, delay-tolerant mMTC applications [4]. Considering the fact that hassle-free devices installation and management are absolutely essential for ubiquitous mMTC penetration, however, it could be argued that the majority of those tweaks are not yet matured enough in terms of markets' adoption to embrace the wave on time. This dictates, to some extents, that mMTC in the first phase of 5G will inevitably run on the 3GPP's well-standardized Long-Term Evolution (LTE) cellular networks whose availability and coverage are found across the continents [5].

Nevertheless, despite being able to provide ubiquitous connections with low end-device complexity and power consumption [6], LTE technology as is may not be suitable for mMTC as suggested by studies from both literature [7], [8] and standardization organizations [9]. The root cause lies in the fact that LTE was originally designed to serve a limited number of human-based devices e.g., smartphones and laptops, all of which generally tolerate high initial access delay but demand speedy data transmission once connection is established. mMTC, on the contrary, possesses 1) several orders of magnitude higher population with an expected density of up to a million MTDs per square kilometer and 2) small data packets but a vast spectrum of delay tolerance due to its various applications [10]. These polar contrasts in nature predict significant challenges for cellular mMTC. Indeed, the consequences of 1) can be seen from [7] where it is demonstrated that in a scenario with tens of thousands MTDs trying to access the Base Station (BS) in a bursty manner e.g., following a power outage, the Random Access CHannel (RACH) of LTE is overloaded and most of the devices cannot obtain access rights before exceeding the allowed number of attempts. On

This work was supported in part by the Japan Society for Promotion of Science (JSPS) under grant number 18K11269.

Anh-Tuan H. Bui, Truong C. Thang, and Anh T. Pham are with the University of Aizu, Japan (corresponding e-mails: buihoanganhtuank56@gmail.com, {thang;pham}@u-aizu.ac.jp).

Chuyen T. Nguyen is with the Hanoi University of Science and Technology, Vietnam (e-mail: chuyen.nguyenthanh@hust.edu.vn).

the other hand, [11] shows that in such overload condition, MTDs of higher priority may suffer from the same delay and success probability penalty as those of lower priority, which essentially violates the requirement of 2).

A. Access Class Barring (ACB)-based solutions

LTE obviously needs significant revamp to be considered as the most suitable access solution for mMTC. The 3GPP is aware of this fact and has adopted the Access Class Barring (ACB) scheme [12] as a baseline access control solution to resolve both issues. The scheme requires an MTD to first generates a random number between [0,1), and if this number is lower than the so-called *barring factor* configured by the BS, the MTD can initiate the Random Access (RA) procedure. RACH overload is thus solved by enforcing low barring factors i.e., restrict the MTDs from initiating access, while setting different factors for different MTD classes handles the prioritization issue. ACB has been shown via both computer simulation [7] and theoretical means [8] to greatly improve access success probability of the MTDs under massive access. The cost, however, is a sharp increase in delay as the barring factors for all classes must be set to very limited values to avoid overload.

There are many proposals to address the delay issue and improve ACB performance. One solution is dynamic ACB where the BS decreases/increases the barring factor in a heuristic manner such as when the average number of collisions in the last three RA slots goes above/below certain thresholds, to greatly reduce service time needed to resolve all MTDs [13]. Various recent works also admit similar adaptive approach but tune the barring factor based on refined estimates of load-indicating quantities, oftentimes the number of backlogged MTDs [14], [15], to achieve even better delay performance. The authors of [16] instead build a fluid model to approximate the "flow" of access requests on RACH and apply non-linear control theory on the model to adapt the barring factor and keep the number of accessing devices per RA slot at the optimal level. Lien *et. al* [17], on the other hand, exploit the fact that MTDs may be covered by multiple BSs in multicell settings and propose a cooperative ACB scheme where MTDs in overlapped regions select their BS according to a predefined strategies based on the BSs' broadcast barring factors. The BSs, knowing such strategies, jointly determine their enforced barring factors so that the maximum number of MTDs accessing any BS is minimized, thus cutting access delay by 30%.

Needless to say, ACB's delay issue is extensively studied. However, many researches in this regard do not comply fully with LTE's current ACB implementation and occasionally overlook a metric important to MTDs - the energy consumption. A prime example of the former is that most dynamic ACB schemes except [18] assume that all backlogged devices are subjected to ACB check even if they have passed the check before, which is not in line with the specs [19]. More importantly, having to power on the radio transceiver to listen and update the dynamic barring factor in every RA slot may quickly exhaust backlogged devices' battery.

On the other hand, to our best knowledge, ACB works that take both *delay* and *prioritization* into consideration are scarce. The authors of [20] use a combination of dynamic access barring and virtual resource allocation to achieve both lower latency and devices classification. Nevertheless, their protocol features a complicated set of parameters that needs to be tuned empirically via means of simulation and thus does not scale well for different scenarios.

To this end, it can be argued that a scalable and practical ACB-based solution to both issues of cellular mMTC is still lacking in literature. Note that there are also PHY-layer approaches e.g., [21] proposes a new preamble detection method to classify devices based on their transmitting power while [22] employs a non-orthogonal multiple access (NOMA)-based strategy to eliminate RA phase and reduce delay. Such solutions show great advantages over ACB-based ones, but also frequently ignore the energy aspect of mMTC and have obvious compatibility issue with LTE systems.

B. Distributed Queueing (DQ)-based solutions

Recently, several studies have promoted the use of a more capable class of Contention Resolution (CR) protocols known as Distributed Queueing (DQ) [23] instead of the RACH's backoff-based one to efficiently handle massive contentions without relying on the ACB. This protocol family divides contending devices into smaller groups and pushes them to the end of a logical "queue". Then in each RA slot, only the head group may leave the queue to reperform access. Thanks to such queueing discipline, DQ-based solutions are very stable under short-term high load and arise as a promising answer to the mMTC access issue. Laya *et. al* [24] pioneer the trend and propose the *Contention Resolution Queue* (CRQ) protocol to tackle a few thousands MTDs attempting to connect to the BS *simultaneously*. Although the CRQ initially outperforms the ACB, its delay degrades rapidly as the population size increases due to the inappropriate employed division rule first disclosed in [25]. This drawback is then amended by the authors of [26] where an adaptive division method aided by a PHY-layer estimation strategy is utilized to resolve up to ten thousands simultaneous MTDs with good access delay. Our previous work in [27], on the other hand, focuses on a more practical 3GPP-referred scenario where 30,000 MTDs try to access in a bursty manner, and proposes the *Free Access Distributed Queue* (FADQ) protocol. Thanks to the free access rule that lets MTDs initiate RA procedure as soon as they need and a MAC-layer estimation method, FADQ achieves good delay performance and is fully compatible with LTE specifications. A more recent DQ-based access solution for mMTC is found in [28] where Lee *et. al* employ a division rule identical to the CRQ protocol but suggest that the head and several subsequent groups retransmit in the same RA slot using different subsets of preambles. Although their scheme does not solve the root problem of CRQ i.e., inappropriate division rule, it does indeed partly mitigates the consequent of such problem to significantly shorten CRQ's access delay. Nevertheless, none of the aforementioned works supports access prioritization.

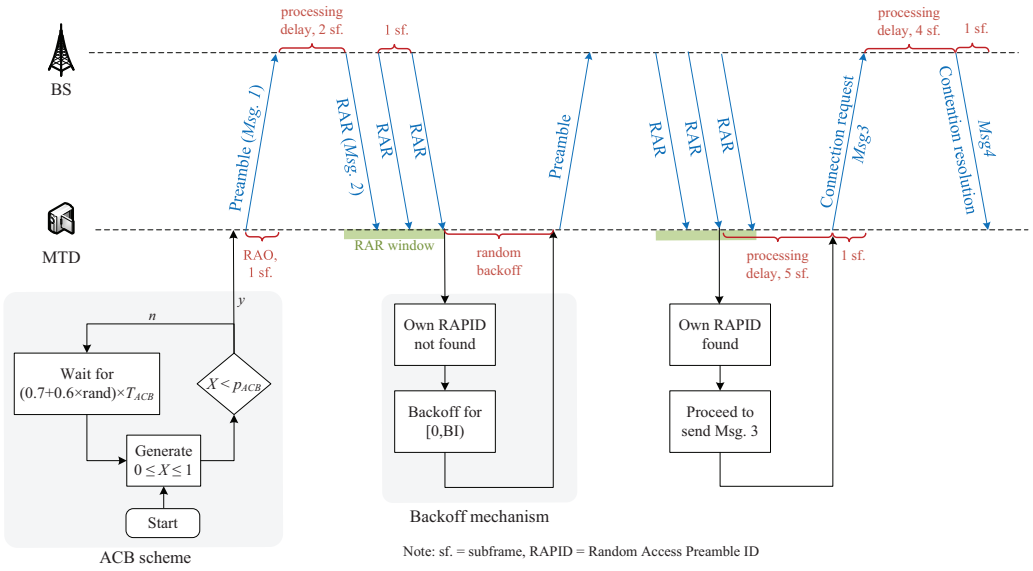

Fig. 1. LTE contention-based random access procedure with ACB activated

C. Motivations of the work and contributions

It is clear, to the best of our knowledge, that there is currently no work that features a complete DQ-based solution to both major issues of cellular mMTC i.e., massive access and devices prioritization. Motivated to fully bring out the potential, in this paper, we aim to introduce a comprehensive DQ-based framework to efficiently and practically support cellular mMTC. This can be considered as a natural yet significant extension of our previous works [26], [27], and the contributions here are outlined as follows.

(a) We propose a DQ-based CR protocol assisted by an existing MAC-layer estimation technique to practically and effectively resolve contentions between MTDs in massive access scenario. Our protocol imposes minimal impacts on current LTE air interface.

(b) A novel prioritization method to differentiate between device classes during RA phase in a dynamic manner is also proposed. By exploiting congestion information from the DQ-based CR process, the proposed method does not incur any additional signaling between MTDs and the BS.

(c) Performance of the proposed framework, mainly in terms of access delay and energy consumption, is evaluated via means of computer simulation and compared with that of ACB schemes to demonstrate ours' effectiveness. Here, an mMTC-oriented traffic model combining both the spontaneous *uniform* and bursty *Beta* access patterns [9] is employed to closely describe behaviors of MTDs in practice.

The rest of this paper is organized as follows. In the next section, the LTE RA procedure as well as the operation of ACB scheme are reviewed. The proposed framework, which includes our DQ-based CR protocol and access prioritization method, is then described in section III. Simulation scenarios and parameters, as well as corresponding results and discussions are provided in section IV. Finally, section V concludes the paper.

II. LTE RA PROCEDURE AND ACB SCHEME

The LTE RA procedure is invoked by a device in five situations (mainly ii), iii), and v) for MTDs): i) the device's initial association with the BS, ii) the device needs to receive/transmit new data but is not synchronized, iii) the device needs to transmit new data but is currently not assigned uplink resource to issue a scheduling request, iv) handover, and v) after a radio-link failure [29]. The procedure is a four-message handshake between the MTD and BS (see Fig. 1) as follows.

- *Msg. 1, RA preamble*: The MTD randomly selects and send one among K orthogonal preambles to the BS in the nearest Random Access Opportunity (RAO). In this step, multiple MTDs may choose the same preamble and cause a preamble collision that may or may not be detected at the BS. In this paper, preamble collisions are assumed to be always detectable [9].
- *Msg. 2, Random Access Response (RAR)*: 2 subframes after its preamble transmission, the MTD starts monitoring the downlink during a window of length W_{RAR} subframes to obtain RAR messages signifying "identities" (IDs) of successfully decoded preambles. Each RAR consumes one subframe in the window and may carry up to N_{RAR} IDs. If the MTD does not find its preamble's ID in any RARs, it backoffs before attempting a preamble retransmission. Otherwise an uplink Msg. 3 is scheduled based on the resource grant and timing instruction in the relevant RAR. Note that the backoff duration is randomly chosen between $[0, BI)$ where BI is the backoff indicator optionally included in the RARs.
- *Msg. 3, Connection request*: Using the reserved uplink resource obtained in Msg. 2, the MTD send a *Connection request* message containing the identifier of itself and reason of access to the BS. This message is protected by Hybrid Automatic Repeat Request (HARQ).
- *Msg. 4, Contention resolution*: When the BS correctly receives Msg. 3 from the MTD, it echoes back the decoded

device's identifier via a *Contention resolution* message as an acknowledgment. The RA procedure is considered successful upon correct reception of this message at the MTD's side. Msg. 4 is also protected by HARQ.

Additionally, each MTD keeps track of the number of its transmitted preambles. If this number exceeds a threshold set by the BS i.e., *preambleTransMax*, the device terminates its RA procedure and gets temporarily "blocked" from accessing the network.

It should be noted that the pattern by which the MTDs become active i.e., "arrive" at the system, and initiate this procedure depends largely on the circumstance. To account for different scenarios, the 3GPP defines two of such patterns, respectively named *uniform* and *Beta*. The former describes normal network usages where the devices arrive in a non-synchronized manner i.e., the time instance at which an individual MTD arrives at the system is a continuous random variable (r.v.) whose probability distribution function (PDF) follows the time-limited uniform distribution. On the other hand, the latter corresponds to extreme scenarios where the MTDs arrive in a highly-synchronized manner e.g., in an attempt to report data related to an external event. Here, the arrival time instance of a device can be modeled as a continuous r.v. whose PDF conforms to the time-limited Beta distribution. The two distributions are given as [9]

$$p(t) = \begin{cases} \dfrac{1}{T}, & \text{Uniform distribution} \\ \dfrac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}B(\alpha,\beta)}, & \text{Beta}(\alpha,\beta)\text{ distribution} \end{cases},$$

where T denotes the *activation period* within which all arrival time instances must fall i.e., $\int_0^T p(t)dt = 1$, and $B(\alpha,\beta)$ is the beta function with parameters $\alpha,\ \beta > 0$.

According to the RA procedure's description, devices involved in preamble collisions are resolved by a random backoff principle, which is reasonable given the non-synchronized i.e., *uniform* arrival pattern. In the bursty *Beta* setting, however, such simple protocol cannot resolve massive short-term congestion, which results in RACH overload and most MTDs being blocked after exceeding *preambleTransMax* attempts [7]. To address the issue, 3GPP has adopted Access Class Barring (ACB) scheme which forces an active MTD to first compare a randomly generated number between $[0,1)$ with the so-called barring factor p_{ACB}. If the former is smaller, the device can initiate the RA procedure. Otherwise, it waits for a random period between $[0.7,1.3)\times T_{ACB}$, where T_{ACB} is the *mean barring time*, before repeating the step. This helps decouple the arrival time and the instance a device actually initiates RA procedure, which results in a load spreading effect as demonstrated in Fig. 2. It is evident from the figure that while ACB greatly reduces the number of new access attempts per RAO during congestion, it also defers devices outside of the congested period and severely degrades access delay.

Despite its toll, ACB scheme is indispensable in providing network connectivity for the MTDs under massive bursty access scenarios because the RACH's backoff-based CR mechanism alone cannot accomplish the task. To efficiently handle the same situation without relying on the ACB (and sacrificing delay), more robust and stable CR protocols must be considered.

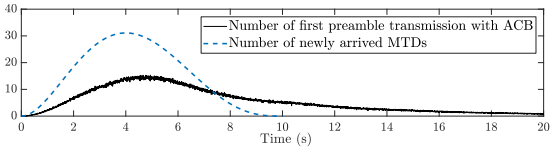

Fig. 2. Traffic spreading effect of ACB scheme with $\{p_{ACB};T_{ACB}\} = \{0.5;4\text{s}\}$ for 30,000 devices arriving according to Beta(3,4) over 10 seconds

III. PROPOSED DQ-BASED FRAMEWORK

A. Estimation Technique

Load estimation is an integral component in the design of adaptive random access protocols capable of offering vastly improved delay performance. In fact, estimation techniques have been extensively studied to shorten tags identification time in Radio Frequency IDenficication (RFID) literature [30], [31], and some have been bridged to cellular MTC context [32]. On the other hand, there also exists novel estimation methods proposed exclusive for cellular mMTC [22], [33]. Many of these solutions, however, modify either the LTE PHY layer or RA procedure and thus, have obvious compatibility issue with existing systems.

To avoid intervening in both LTE air interface and the RA procedure itself, in this paper, we employ a MAC-layer approach similar to Vogt's in [34]. This enables the BS to estimate the number of colliding MTDs in an RAO based on the observed statuses of the preambles as follows. Let us respectively denote by C, S and E the numbers of preambles that are transmitted by more than one (Collision), only one (Singleton) and none (Empty) of the MTDs in an RAO from the BS's perspective. Note that $C+S+E=K$ i.e., total number of preambles. When n_t devices are transmitting in an RAO, the expectations of C, S and E can be respectively derived as

$$\bar{C} = K - \bar{S} - \bar{E}, \quad \bar{S} = n_t\left(1-\frac{1}{K}\right)^{n_t-1}, \quad \bar{E} = K\left(1-\frac{1}{K}\right)^{n_t}. \quad (1)$$

Estimation of the parameter n_t based on the joint distribution of C, S, and E turns out cumbersome due to the lack of closed-forms [14], [34], [35]. However, it is well known that if an r.v. has a small variance, the probability of its taking values far from its mean is proportionally small. In particular, if X is an r.v. with finite mean μ and variance σ^2, then $\mathbb{P}\left[\left|X-\mu\right| \geq m\sigma\right] \leq 1/m^2$ for any $m>0$ (Chebyshev's inequality). Parameter estimation can therefore be done based on distance to the mean instead of exact distributions for r.v.s with small variances. Since C, S, and E have relatively small σ^2 [36], the BS can reasonably obtain an estimate $\hat{n}_t$ of the parameter n_t by searching for the value that minimizes the distance between the actual outcome of the triplet (C,S,E) and their theoretical means,

$$\hat{n}_t = \arg\min_{n_t\in\mathbb{N}}\left\{\left(C-\bar{C}\right)^2+\left(S-\bar{S}\right)^2+\left(E-\bar{E}\right)^2\right\}. \quad (2)$$

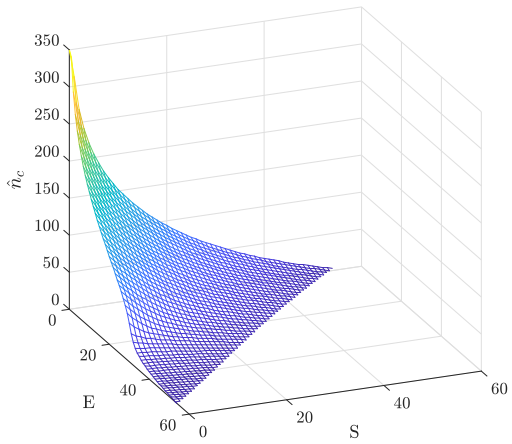

Fig. 3. Obtained $\hat{n}_c$ for different outcomes of (C,S,E), given $K = 54$

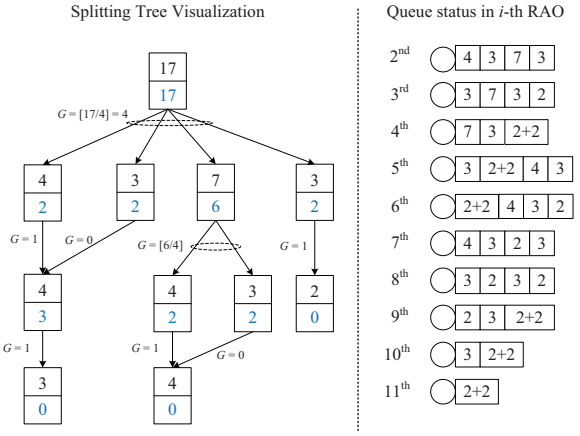

Fig. 4. Operation of proposed DQ-based contention resolution protocol, $K = 4$

In the unlikely event of $C = K$ i.e., all preambles are in collision, $\hat{n}_t$ cannot be found by (2) and is interpolated as the integer multiple of K that is greater and closest to the $\hat{n}_t$ at $C = K - 1$. Finally, the estimated number of colliding MTDs i.e., $\hat{n}_c$, is simply found as $\hat{n}_c = \hat{n}_t - S$. The accuracy of this method is further discussed in section III-D.

Since estimation is executed in every RAO, the speed at which the BS derives $\hat{n}_c$ is also important. We now prove that $\hat{n}_c$ can be derived in real-time as follows. For a certain K, the set of possible outcomes for the triplet (C,S,E) is also finite. Furthermore, given a certain outcome, there is only one $\hat{n}_t$ satisfying (2). This means that $\hat{n}_t$ (and hence, $\hat{n}_c$) for each and every possible triplet in the finite set can be calculated offline, and the BS can construct a static lookup table that maps each (C,S,E) outcome to its corresponding $\hat{n}_c$ solution in advance. To estimate the number of colliding MTDs in an RAO, the BS then simply looks for the entry that matches the actual triplet observed in that RAO and extracts the relevant $\hat{n}_c$. The lookup process incurs negligible delay, and estimation can therefore be done almost instantly. Furthermore, since K is capped at 64 [36], the BS can compute and store all 64 possible tables offline to address all configurations of K.

Fig. 3 depicts such static lookup table in graphical form, given a typical assumption of $K = 54$ [9]. Note that C is omitted due to the constrain of $C + S + E = K$. It is seen from the figure that the BS can always quickly obtain the estimate $\hat{n}_c$ given any actual outcome of the triplet.

B. Proposed DQ-based Contention Resolution Protocol

It is proven in [26] that the poor delay performance of the conventional DQ-based protocol i.e., the CRQ, is caused by its division rule that divides colliding MTDs into groups based solely on their preambles without estimating their actual number. When there are many preamble collisions, each between only a few MTDs, many tiny groups will be created. With a group always consuming a full RAO regardless of its size, the RAOs in which these tiny groups retransmit will be under-utilized, as the number of transmitting MTDs is much lower than what an RAO can handle.

To avoid under-utilization, the number of devices per group must be controlled. One possible solution is to estimate the number of colliding devices and build the division rule around the estimation. This has motivated us to propose a new DQ-based CR protocol aided by the estimation method in III-A to utilize RA resources in an effective manner. The key idea here is that whenever preamble collisions occur in an RAO, all n_c colliding MTDs (regardless of which collisions they are involved in) are randomly divided into G groups. More importantly, G is determined by the BS based on the estimate $\hat{n}_c$ to keep the average size of a group at an optimal level. For convenience, let us denote by r the maximum *expected* number of MTDs that can successfully obtain uplink grants in an RAO, and by $\omega(r)$ the number of MTDs in the RAO so that r can be achieved. $\omega(r)$ is derived in Appendix A, while G is calculated as follows.

- If $\hat{n}_c > \omega(r)$, then $G = \lfloor \hat{n}_c/\omega(r) \rceil$ where $\lfloor \cdot \rceil$ denotes the "round" operator. The rationale is that after division, the average number of MTDs per group will be around $\omega(r)$ and thus, r may be attained when these groups retransmit in the future.
- If $\hat{n}_c \leq \omega(r)$, then $G = 1$ as at this point, any further divisions will cause the average number of MTDs per group to drop too low, which leads to under-utilization.

These G groups are then "pushed" to the end of a logical access queue where in each RAO, only the head group may exit and perform preamble retransmissions. Note that the queue itself does not exist physically, but is realized using two counters named DQ and pDQ. The former is maintained exclusively by the BS to keep track of the queue's "length" i.e., the number of groups that are currently inside the queue. On the other hand, the latter is "distributed" at each individual MTD to informs the device about its current "position" inside the queue e.g., those whose $pDQ = 0$ are at the queue's head and may transmit their preambles in the RAO. These counters encode all relevant information on the queue, and are updated after each RAO as follows.

For DQ (at the BS):

- If $DQ > 0$ i.e., a contention session is going on, then $DQ = DQ - 1$ due to removal of the head entry.

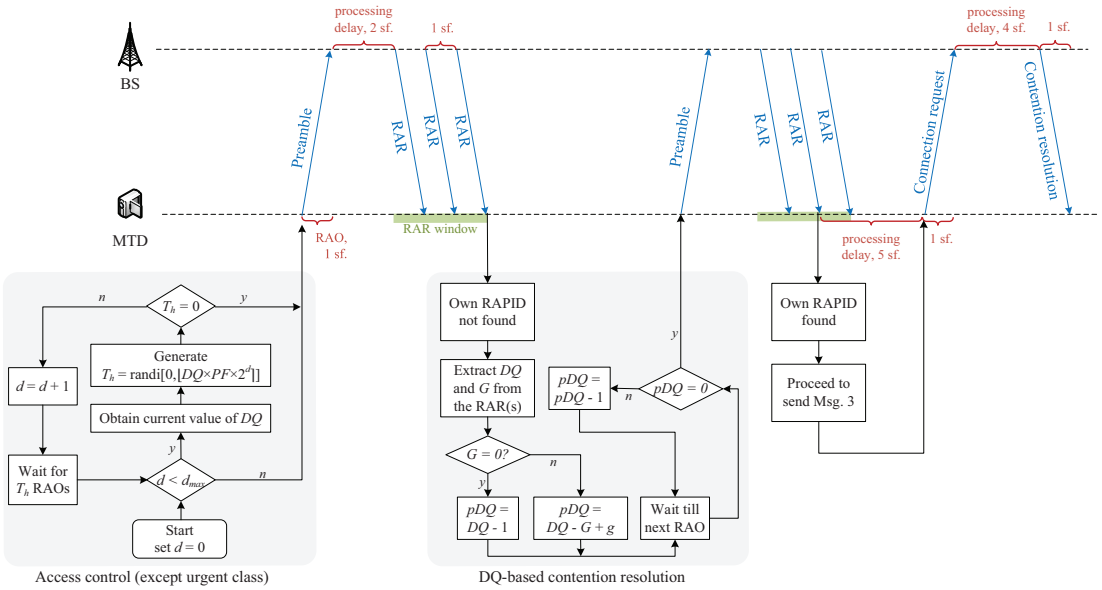

Fig. 5. Random access procedure for the DQ-based framework

- If preamble collisions occur, $DQ = DQ + G$ to reflect the addition of G groups of MTDs to the queue's end.

For pDQ (at individual MTD):

- If the MTD is waiting in the queue i.e., $pDQ > 0$, then $pDQ = pDQ - 1$ due to removal of the head entry.
- If the MTD is involved in a preamble collision, it selects a random integer g between $[0, G-1]$ and sets its $pDQ = DQ - G + g$ to indicate that it has chosen the g-th group and re-entered the queue from the end, respectively. Note that the most updated values of G and DQ are assumed to be included in the corresponding RARs by the BS.

At this point, we newly notice that when the number of MTDs per group drops too low e.g., during post-congestion, it may be more beneficial to merge some of them together instead of letting them shrink monotonically by setting $G = 1$ as in our previous work [27]. To realize this new improvement, we assume that the BS keeps an estimate of the tail group's size i.e., $\hat{n}_e$. Whenever preamble collisions happen in an RAO, colliding MTDs will merge with that tail group instead of forming new group(s) if the estimated size after merging i.e., $\hat{n}_e + \hat{n}_c$, does not exceed the optimal level $\omega(r)$. Otherwise, $G \geq 1$ group(s) are created at the queue's end following the usual formula. Accordingly, BS and MTDs must apply additional update rules as follows

For BS:

- If $\hat{n}_e + \hat{n}_c \leq \omega(r)$, group merging occurs. In this case, $G = 0$ i.e., no new group is created, and the estimated tail group's size is updated as $\hat{n}_e = \hat{n}_e + \hat{n}_c$. Otherwise, group merging is not allowed and $G \geq 1$ groups will be created at the queue's end as usual. $\hat{n}_e$ is thus updated as $\hat{n}_c / G$, which is the estimated average number of MTDs per newly created group.

For individual MTD:

- If preamble collision occurs and $G = 0$, then $pDQ = DQ - 1$ to merge with the tail group.

Fig. 4 explains how contentions between MTDs are handled, given $K = 4$. For demonstration purpose, let us assume that the BS knows exactly the number of colliding MTDs in an RAO as well as the number of MTDs in each created group, and that $W_{RAR} \times N_{RAR}$ is big enough so that $\omega(r) = K$ (see Appendix A). Each rectangle here represents an RAO, while the upper and lower numbers inside respectively denote the number of transmitting i.e., n_t, and colliding MTDs i.e., n_c, in that RAO. In the very first RAO when the queue is still empty, 17 MTDs transmit and cause collisions in all 4 preambles i.e., $n_t = n_c = 17$. The BS thus randomly divides these MTDs into $G = \lfloor \hat{n}_c / \omega(r) \rceil = 4$ groups of respective sizes 4, 3, 7, 3 and updates a record of tail group size i.e., $n_e = 3$. Note how the average number of MTDs per group is close to the optimal of $\omega(r) = K = 4$. The first group of 4 MTDs leave the queue and retransmit in the 2nd RAO where two are involved in a collision i.e., $n_c = 2$. Since $n_c + n_e = 5 > \omega(r)$ and $n_c < \omega(r)$, the BS 1) decides that group merging should not happen and 2) sets $G = 1$ to avoid over-division, respectively. In addition, it also updates $n_e = 2$. The two remaining MTDs then simply rejoin the queue as a single group at the end to retry later (in the 6th RAO). The second group of 3 MTDs retransmit in the 3rd RAO and two also collide. In this case, the BS sees that $n_c + n_e = 4 \leq \omega(r)$ and set $G = 0$ in the RARs to instruct colliding devices to perform merging. The two MTDs in question are thus merged with the tail group, and n_e is also updated accordingly i.e., $n_e = 2 + 2 = 4$. These processes continue until the end of 11th RAO by which all MTDs have been resolved.

Note that as a CR protocol, our proposal here is only for resolving contentions as they arise between MTDs, and does not define how new devices arriving in the middle of a contention process are treated. The latter task is, in fact, the job of access control schemes (such as the ACB). To finalize our DQ-based access framework design and realize differentiation between MTD classes during RA phase, we now proceed to

describe our novel access control method which will also play the role of prioritization.

C. Proposed Prioritization Method

It is seen that in our DQ-based CR process, the queue's length DQ can be properly used as an indicator of the system's congestion level. When the system is stressed, DQ keeps increasing as new groups are consistently created and added to the queue in each RAOs due to heavy contentions. On the other hand, DQ of a lightly-loaded system tends to stay small since preamble collisions rarely occur and the devices are granted access almost instantly. This key observation allows us to propose a novel dynamic access control scheme to achieve efficient prioritization between MTD classes without introducing additional signaling overhead. In order to do so, our proposal defers RA procedure initiation of different device classes by different amounts of time based on not only the priorities of the classes, but also current value of DQ.

Specifically, in our scheme, a device who is about to initiate the RA procedure in an RAO must first generate a non-negative random number T_h. If the generated $T_h = 0$, the MTD can initiate its RA procedure as planned. Otherwise, it has to wait for T_h RAOs before re-calculating T_h. Note that in the proposed formula below, DQ must be updated to its very current value every time T_h is to be (re)generated.

$$T_h = \text{randi}\left[0,\ \left\lfloor DQ \times PF \times 2^d \right\rceil \right], \tag{3}$$

where randi$[a,b]$ is a function that generates a random integer in the range of $[a,b]$. PF is a "prioritization factor" that can take any non-negative value, and is used to differentiate between device classes. MTDs of a higher PF belongs to a lower-priority class as they must hold their RA procedure longer. Note that the urgent class has $PF = 0$ i.e., its MTDs can immediately initiate access as soon as they arrive. Additionally, each device keeps track of the number of times its RA procedure has been deferred via an integer counter $d \geq 0$. The term 2^d can then be understood as an exponential-growing factor whose rationale is similar to that of the well-known Binary Exponential Backoff (BEB) algorithm. To avoid the case where an MTD has to indefinitely defer its RA procedure, a common limit d_{max} is set for all classes, and devices whose $d \geq d_{max}$ is allowed to initiate access without any further deferment. The RA procedure, slightly modified to accommodate our whole framework is then depicted in Fig. 5. Note that with our prioritization scheme, our derivation of G in III-B is no longer optimal since the number of transmitting devices in an RAO is now the total of not only the number of MTDs in the head group (which is already at the optimal level), but also the number of new MTDs that are allowed to initiate its RA procedure in the same RAO. In other words, given the proposed prioritization scheme, our calculation of G only represents a *best-effort* solution in keeping the number of MTDs per RAO at the optimal level.

Since DQ, PF and d are involved in the calculation of T_h, MTDs must obtain these parameters. PF for different MTDs classes and d_{max} are static, and can be trivially broadcast by the BS as part of the system information block (SIB2). On the other hand, DQ is a fast-changing parameter that cannot be updated frequently via broadcast message, as the refreshing interval of SIB2 is relatively slow (typically 160ms) [37]. However, in our proposed DQ-based CR protocol, the current value of DQ is readily available in the RARs. Thus, non-urgent devices who need to generate their T_h may easily get the most updated value of DQ by obtaining any RAR corresponding to the nearest past RAO without requiring extra information exchange with the BS.

D. Other Considerations

Here, we consider some practical issues when implementing the proposed DQ-based framework into an actual LTE system. These includes the situations in which a) the time spacing between consecutive RAOs is too short such that a new RAO comes before the finish of current RAR window, and b) MTDs' preamble transmissions fail due to reasons not related to collision. Also, the impact of estimation accuracy on performance of our framework is briefly discussed.

In LTE systems, subframes allocated for random access purposes i.e., RAOs, are allocated periodically on time domain. The periodicity of RAOs is broadcast by the BS via the parameter *PRACHConfIndex*, and may vary from one RAO per 20 subframes to an RAO every subframe [38]. Issue a) comes from the fact that if the RAO periodicity is short enough, a new RAO may come before devices of the current RAO finish capturing the RAR window. This causes RAOs wastage because all devices involved in the DQ-based contention process cannot update their counters until the RAR window is over to ensure synchronous operation of the queue. All new RAOs that come before the RAR window's completion are thus considered unusable by the system.

The top of Fig. 6 depicts this issue given that *PRACH-ConfIndex* = 6 i.e., one RAO per 5 subframes, and the RAR window's length is $W_{RAR} = 5$ subframes. For brevity's sake, we refer to the location of a subframe as (a,b) where a and b are the corresponding frame type ("o" for odd and "e" for even) and subframe number, respectively. Assuming that MTDs of the head group send their preambles in the RAO at (o,1), it is then evident that the RAO at (o,6) takes place during the window where these devices are still busy capturing RARs and thus, become unusable for the whole system. This in turn means that only the pair of RAOs at (o,1) and (e,1) can be used for the contention process, which result in only 50% utilization of available RAOs.

To avoid such wastage, we employ multiple parallel queues as in our previous work [26] so that RAOs unusable by a queue are exploited by other queues. For the configuration of Fig. 6, two parallel queues are formed. The first one is assigned the RAOs pair at (o,1) and (e,1) while the second uses those at (o,6) and (e,6). The assignment ensures that from a queue's perspective, RAR window for an RAO always finishes before the occurrence of the next RAO, as illustrated by the figure where RAR windows of (o,1) and (o,6) end well before the coming of corresponding next RAOs respectively located at (e,1) and (e,6). All RAOs can thus be utilized, and this idea

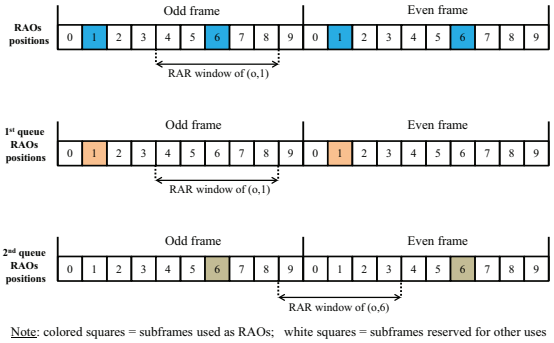

Fig. 6. Multiple parallel distributed queues concept

extends naturally to other configurations as well. We assume that the number of queues and RAOs location of each are broadcast as part of system information. Upon joining the network, each MTD randomly selects a queue to associate with and all of its RAO-related timings e.g., counters adjustment and holding duration of T_h, are then explicitly tied to RAOs seen by that queue.

Issue b), on the other hand, arises in accordance with 3GPP simulation assumptions which insist that a *singleton preamble* is detected with probability $\left(1-1/e^{i}\right)$ if it is the i-th attempt by the device to account for the radio channel and power ramping effect. Furthermore, even if the singleton preamble is correctly detected, the limited number of uplink grants during an RAR window may result in the MTD not being granted resource for Msg. 3 transmission. In this paper, we assume that MTDs whose preamble transmissions fail due to these reasons are aware of such fact and behave similarly to those who are yet to initiate RA procedure. That is, the device is subjected to the deferment process of T_h if its $d < d_{max}$ or retransmit in the nearest RAO seen by its chosen queue otherwise.

Lastly, all estimation methods, including our employed one, may produce estimates that deviate from true values up to certain degrees. Furthermore, since our method relies on the *observed* statuses of the preambles, issue b) which causes singleton preambles to be detected as empty ones by the BS further lowers the estimation's accuracy. However, we have verified via computer simulations[1] that knowing the exact number of colliding devices in an RAO (perfect estimation) does not significantly affect performance of our framework, with the variations being in orders of less than 3%. This is because the estimation error is also divided by $\omega(r)$ during the process of determining the number of groups G, which partly mitigates its impact on system performance. It is thus advised to keep in mind that although such flawed estimation technique is actually implemented in our simulation, the obtained performance is not that far away from the ideal case.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we detail the setups used to test the performance of our framework and other schemes of interest. A simple energy model is also introduced to qualitatively measure the total amount of energy consumed by the (most likely battery-limited) MTDs. System-level simulation using basic MATLAB programming[1] are then performed to demonstrate the effectiveness of our solution. Note that although there exists various DQ-based schemes, most of them focus on simultaneous arrivals traffic model and none supports access prioritization. Extension of these works to our multi-class settings is non-trivial. This has driven us to compare our framework with ACB-based solutions, which readily supports prioritization, for the sake of a fair comparison.

[1]Source codes available at: www.u-aizu.ac.jp/labs/ce-cc/2019.TVT-Codes.zip

A. Simulation Setup

Simulation parameters are summarized in Table I, which mostly agrees with 3GPP's setup in [9]. A major difference is our employed traffic model that considers both the bursty Beta and the sporadic uniform access patterns to reflect the coexistence of highly-synchronized and non-synchronized MTDs in the network. Specifically, a ratio x of the population is assumed to arrive following Beta distribution while the remaining $(1-x)$ does so according to the uniform distribution, both over an *activation period* of 10 seconds. Three ***scenarios*** of $x = 0.25$, 0.5, and 0.75 are then defined in our simulation. We want to emphasize that although all scenarios are required to facilitate discussion, those with higher x i.e., more bursty traffic, are of greater interest in practice. Also, access control parameters for ACB and ours in III-C are not detailed in Table I but revealed later for convenience.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Number of MTDs	$N = 40{,}000$
Arrival distribution	Ratio x: Beta(3,4) over 10s Ratio $(1-x)$: Uniform over 10s
PRACH configuration index	$PRACHConfIndex = 6$
Subframe length	$t_{sf} = 1$ ms
Available preambles for contention-based random access	$K = 54$
Maximum number of preamble transmissions	$preambleTransMax = 10$
RAR window size	$W_{RAR} = 5$ subframes
Maximum number of uplink grants per RAR	$N_{RAR} = 3$
Preamble detection probability for the i-th preamble transmission	$P_d = 1 - \frac{1}{e^i}$
B-ACB Backoff Indicator	$BI = 960$ ms
Retransmission probability for Msg 3 and Msg 4	0.1
Maximum number of Msg 3 and Msg 4 HARQ transmissions	5
Round-trip time of Msg 3 (Msg 4)	8 (5) subframes

Our employed energy model assumes that a device is always in either "Transmitting" (Tx), "Receiving" (Rx), or "Idle" states once it arrives. Power consumption in each states is respectively set to $P_1 = 50$mW, $P_2 = 50$mW and $P_3 = 0.025$mW as in [39]. An MTD is in

- Tx state: in a subframe where it sends either a preamble, a Msg. 3, or a Msg. 4 HARQ feedback to the BS.
- Rx state: in a subframe where it captures either an RAR, a Msg. 3 HARQ feedback, a Msg. 4, or an updated barring factor (in case of dynamic ACB) from the BS.
- Idle state: otherwise.

TABLE II
CONTENTION RESOLUTION PERFORMANCE RESULTS

	$x = 0.25$		$x = 0.5$		$x = 0.75$	
	B-ACB	***Proposal***	***B-ACB***	***Proposal***	***B-ACB***	***Proposal***
Blocking probability, P_b	3.14%	0.34%	3.73%	0.24%	3.95%	0.18%
Access delay, $\mathbb{E}[D]$	3255.8 ms	2354.2ms	3808.9 ms	2767.8 ms	4435.4 ms	3331.9 ms
Energy consumption	39.385 J	37.969 J	39.8204 J	37.703 J	40.154 J	37.913 J
Number of preamble transmissions, N_{PT}	3.107	2.933	3.127	2.878	3.118	2.852

The total amount of consumed energy is then calculated by $t_{sf} \times (P_1 \times n_{tx} + P_2 \times n_{rx} + P_3 \times n_{idle})$ where t_{sf}, n_{tx}, n_{rx}, n_{idle} respectively denote the subframe length, the number of Tx, Rx, and idle subframes.

Note that when an MTD wants to obtain current DQ, it simply captures one RAR from the nearest window since DQ is included in every RARs. Otherwise the device is looking for an RAR as part of the CR process and might need to capture multiple RARs until it either finds a relevant one or gives up after the window is over [12]. In the former case, the number of RARs it must capture, which is also the number of needed Rx subframes, equals the order of its RAR among W_{RAR} ones. In the latter case, the MTD has to capture all W_{RAR} RARs without finding its RAR.

As seen, the model is very simple in the sense that it defines the states solely based on the device RF module's activity and admits a fixed power consumption level at each state. This ignores various factors that may cause the actual expense to vary greatly e.g., the MTD's chosen Modulation and Coding Scheme (MCS), the power ramping mentioned in section III-D (which is a typical example of LTE's open-loop power control), and myriad signal processing going on when the MTD is neither transmitting nor receiving. However, since we are mainly interested in MAC protocol design, such model serves as a good starting point to quantify the tradeoff between delay/energy brings about by the protocols of interest. Similar assumptions are also spotted in MAC-layer researches [40] although their exact power figures may differ from ours. Enthusiasts are referred to [41] for an in-depth power model built on realistic measurements from an LTE dongle.

For the sake of brevity, we henceforth refer to devices that successfully gain access before exceeding *PreambleTransMax* as *successful* MTDs. Performance of the frameworks is then assessed via four main metrics:

1) Blocking probability P_b: the ratio of the number of blocked devices to N.
2) Average access delay $\mathbb{E}[D]$: the average duration from when a successful MTD arrives until it correctly receives Msg. 4.
3) Total energy consumption: the total amount of energy consumed by all successful MTDs during RA phase.
4) Number of preamble transmission N_{PT}: the average number of times a successful MTD has to transmit a preamble before it succeeds.

B. Contention Resolution Performance

In this precursor part, we ignore classification/access control and set $PF = 0$ universally to see how well our DQ-based CR protocol handles massive access. Since the LTE backoff-based counterpart alone cannot manage the same situation, we additionally employ B-ACB just to provide a meaningful reference. The barring parameters are chosen heuristically to keep $P_b \leq 5\%$ and $\mathbb{E}[D]$ at reasonable levels [7]. In particular, $\{p_{ACB}; T_{ACB}\} = \{0.65; 4s\}$, $\{0.6; 4s\}$, and $\{0.55; 4s\}$ for $x = 0.25$, 0.5, and 0.75, respectively. Note that performance in multi-class settings, which is of main interest, will be addressed later where a dynamic ACB scheme is also considered.

Simulation results are then shown in Table II. The proposed CR mechanism clearly reduces both blocking probability, access delay, and energy consumption compared to B-ACB. The focus in this section is, however, not such comparison but to scrutinize our framework's reactions to changes in Beta traffic ratio x to facilitate future understanding as follows.

1) *In terms of access delay*, higher x leaves negative impacts due to a proportional increase in number of competing devices during peak period. Consequently, more groups are created per RAO during the period, which causes the queue to extend rapidly and prolong queuing delay. To back the argument up, we have plotted in Fig. 7 the temporal evolution of the queue's length DQ. The queue is clearly longest in the third scenario. This figure also shows that despite the congestion, our framework can keep the number of granted MTDs consistently close to the system's limit of $W_{RAR} \times N_{RAR} = 15$, which implies near optimal delay performance.

2) N_{PT} *and hence,* P_b drop when x increases. This is counterintuitive and can be explained by noting that MTDs of such cases are more likely to collide on the first try but may finish within just a few next attempts which occur much later (due to high DQ) and may fall outside the activation period. On the contrary, devices of low-x scenarios have slightly better success chance on their first try but retries, if any, are timed shortly afterward (due to low DQ) and fall within activation period where there are new arrivals. Such retries experience relatively less success chance, netting a higher number of retransmissions that offsets the first try advantage. An obvious tradeoff for this is a much lower $\mathbb{E}[D]$.

3) *Total energy consumption* seems to react inconsistently to variations in x due to a canceling effect between N_{PT} and $\mathbb{E}[D]$. A low N_{PT} saves an appreciable amount of energy because it reduces both the time a device spends transmitting and capturing whole RAR windows. However, a prolonged

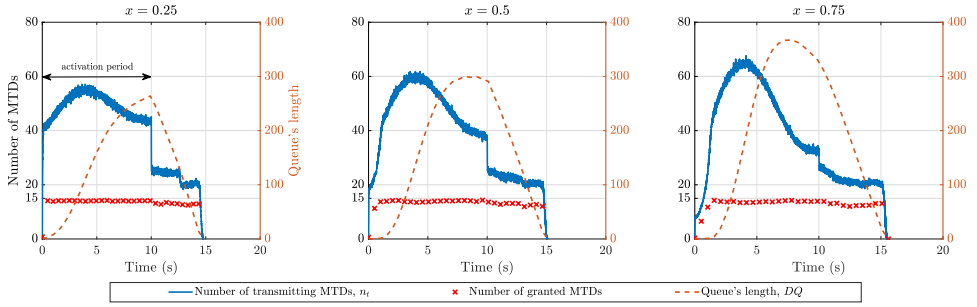

Fig. 7. System evolution over time for three scenarios of Beta traffic ratio $x = 0.25$, 0.5, and 0.75

TABLE III
ACCESS CONTROL PARAMETERS CONFIGURATIONS FOR IV-C

	B-ACB p_{ACB}[†]	***D-ACB*** T_{ACB}	***Proposal*** PF & d_{max}
$x = 0.25$	1, 0.6, 0.4	0s, 0.5s, 10s	d_{max} = 2 or 3 First PF set 0, 1, 2 Second PF set 0, 1.5, 3
$x = 0.5$	0.85, 0.4, 0.2	1.2s, 6s, 12s	
$x = 0.75$	0.65, 0.3, 0.15	2s, 12s, 18s	

Values from left to right of a triplet are for class 0, 1, and 2, respectively.
[†] T_{ACB} of B-ACB is 4s for all classes.

$\mathbb{E}[D]$ can overshadow that gain because an MTD consumes non-negligible energy in idle mode. As an example, there is on average 0.055 fewer attempts and 413.6ms more delay when x moves from 0.25 to 0.5, which roughly corresponds to a net energy reduction of $0.055 \times 6 \times 50 - 413.6 \times 0.025 = 6.16\mu J$ per device. When x changes from 0.5 to 0.75, this number becomes negative and causes an increase in consumption level.

C. Framework Performance

We are now in a position to analyze the proposed framework under multi-class settings. In this main section, we assume that Beta MTDs belong to an urgent class 0 while others are delay-tolerant [42]. For more insights, the latter is further divided into two subclasses i.e., class 1 and 2, of equal size but different priorities. Performance of our framework is then compared with both B-ACB and a Dynamic ACB (D-ACB) scheme in [18]. Note that [18] is chosen because it complies with 3GPP specifications in the sense that MTDs who passed their ACB checks are not subjected to the check afterwards. To stabilize the RACH, [18] dynamically adjusts the barring factor based on the number of devices in backoff state that are expected to retransmit in current RAO and broadcast it via an SIB2 update every 80ms.

Since [18] does not specify how to differentiate between device classes and the dynamic barring factor p_{ACB} must be universal, we assume that classification is realized via class-specific T_{ACB}. Prioritization settings are then detailed in Table III where barring parameters for ACB-based solutions are again selected heuristically to ensure blocking probabilities of less than 5% while maintaining a low access delay for class 0. Note that all selections are based on the premise of $BI = 960$ms and 20ms for B-ACB and D-ACB [18], respectively.

As indicated by eq. (3), our framework is tunable via d_{max} and PF. The former is an exponential factor used for adjustments on a macro scale. The latter, on the contrary, is linear and provides a more granular tuning option. To first study the effect of macro-tuning on comparative performance, we fix PF to the first set and vary d_{max} between 2 and 3. Corresponding delay results are outlined in Table IV as the first two top values in entries corresponding to our proposal and plotted in Fig. 8 for better visualization.

1) Comparison with ACB: In terms of *access delay*, Fig. 8 shows that the DQ-based framework offers performance levels that range from comparable to significantly better than ACB counterparts. In particular, $\mathbb{E}[D]$ are grossly lower in ours than in B-ACB regardless of classes. D-ACB, on the other hand, manages to sometime pull ahead in terms of class 0 delay in the first scenario but still lags behind otherwise, especially in the third scenario where large margins in our favor are observed. The delay advantage, coupled with a lower number of transmissions N_{PT} (see Fig. 9), has validated the effectiveness of using DQ to realize dynamic prioritization.

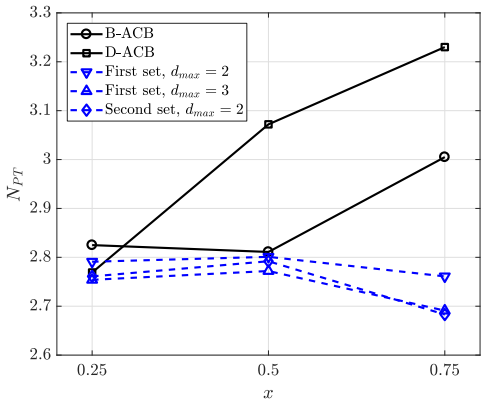

Fig. 9. Average number of preamble transmissions, N_{PT}

Displayed in Fig. 10 is the *blocking probability*. Owing to its efficient DQ-based CR module, the proposed framework achieves close-to-zero P_b regardless of settings to surpass both B-ACB and D-ACB. It is also noted that P_b performance of ACB-based solutions are very sensitive to changes in barring parameters, which limits the freedom in configuring prioritization profiles.

Our *total energy consumption*, as depicted in Fig. 11, is remarkably lower than ACB-based solutions given the most

TABLE IV
FRAMEWORK DELAY PERFORMANCE RESULTS

	$x = 0.25$			$x = 0.5$			$x = 0.75$		
	B-ACB	***D-ACB***	***Proposal***	***B-ACB***	***D-ACB***	***Proposal***	***B-ACB***	***D-ACB***	***Proposal***
$\mathbb{E}[D]$, class 0	1103.4 ms	692.4 ms	761.4 ms 371.6 ms 585.2 ms	1901.7 ms	1925.7 ms	1482.8 ms 1103.8 ms 1321.4 ms	3381.9 ms	2917.7 ms	2608.1 ms 2402.7 ms 2526.2 ms
$\mathbb{E}[D]$, class 1	3623.4 ms	2819.5 ms	2391.8 ms 2517.2 ms 2417.0 ms	6821.5 ms	3788.0 ms	3277.1 ms 3808.0 ms 3491.7 ms	10163.4 ms	6646.5 ms	4737.1 ms 6380.4 ms 5533.4 ms
$\mathbb{E}[D]$, class 2	6870.5 ms	4678.9 ms	3871.6 ms 4562.9 ms 4157.7 ms	16704.2 ms	6454.5 ms	4990.2 ms 5778.4 ms 5272.1 ms	23375.3 ms	9867.0 ms	6776.5 ms 8624.5 ms 8054.3 ms

Values from top to bottom in an entry of our proposal are results obtained by setting $\{PF, d_{max}\}$ to {first set, 2}, {first set, 3}, and {second set, 2}, respectively.

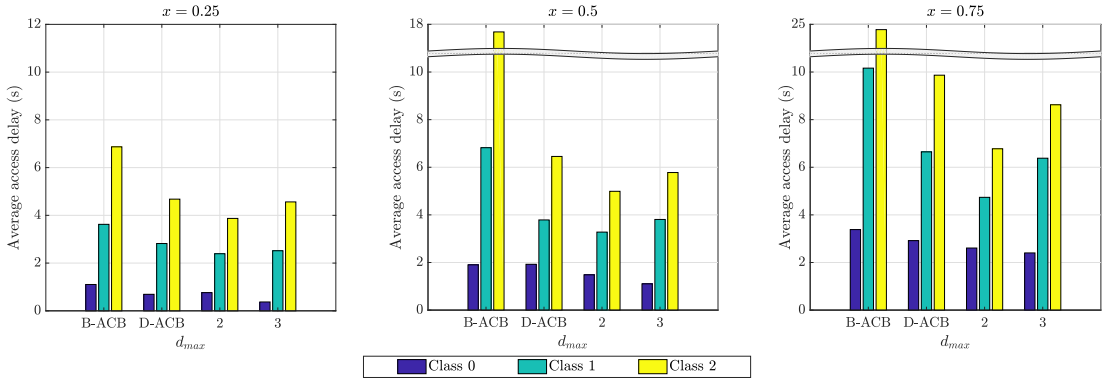

Fig. 8. Average access delay $\mathbb{E}[D]$ of MTD classes in the three scenarios, assuming the first PF set

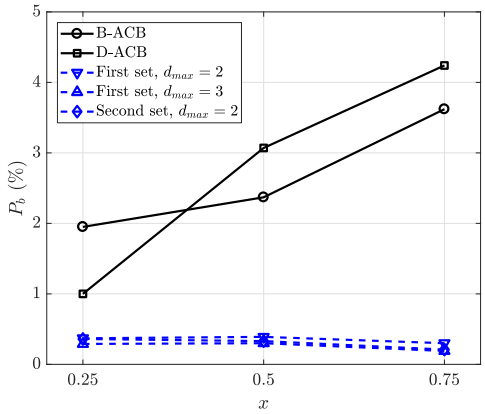

Fig. 10. Overall blocking probability, P_b

interested scenario of $x = 0.75$. In multi-class settings, the consumption level is a complex result of interaction between $\mathbb{E}[D]$, N_{PT} (see section IV-B), and additionally, for dynamic schemes, the number of times an MTD has to capture the dynamic factor e.g., DQ or p_{ACB}. The shortcomings of ACB then become intuitive: B-ACB is penalized by atrocious $\mathbb{E}[D]$ of non-urgent MTDs whereas D-ACB forces the whole population to capture new p_{ACB}. The advantage of our framework, meanwhile, is justified by both lower $\mathbb{E}[D]$ and N_{PT}, and the fact that only the lesser non-urgent portion have to capture DQ. Scenarios with more non-urgent devices i.e., lower x, are of course naturally against us, but a consumption level below both ACB counterparts (except at the least interested $x = 0.25$ scenario) can still be obtained by *tuning* as shown later.

2) Effect of macro-tuning: It is seen from Fig. 10 that tuning d_{max} has no noticeable effect on the low blocking probability P_b of our framework. In terms of $\mathbb{E}[D]$, however, macro-tuning provides a means to switch between very different classification profiles. A higher d_{max} means a more obvious classification effect because non-urgent MTDs are scattered more greatly to lessen overall contentions (suggested by a lower N_{PT} in Fig. 9). As a result, delay of urgent devices is improved while that of non-urgent ones is degraded as confirmed by Fig. 8. How high should d_{max} be will therefore correlate with how important class 0 is to network operators but at the same time, d_{max} should not be too high to avoid extreme delay. According to Fig. 8, setting $d_{max} = 3$ will secure better delay for class 0 while still maintaining comparable performance for class 1 & 2 compared to D-ACB.

Energy wise, boosting d_{max} generally raises the usage level (see Fig. 11), which is obvious since d_{max} is the upper limit on the number of times a non-urgent device has to capture DQ. Furthermore, (3) implies that non-urgent MTDs are almost guaranteed to undergo all d_{max} deferments in massive access. To prolong MTDs' overall lifetime, d_{max} should be kept small. Fig. 11 shows that our framework consumes less energy than D-ACB with d_{max} as low as 2. A dilemma thus arises: to surpass D-ACB in terms of class 0 delay while keeping a comparable performance for class 1 & 2, d_{max} should be 3 but to outperform D-ACB in energy domain requires d_{max} to be kept at 2. This is where fine-tuning becomes necessary.

3) Further improvement via fine-tuning: Now that effects of macro-tuning are revealed, we fix $d_{max} = 2$ for a good energy consumption level, then further improve class 0 delay via PF.

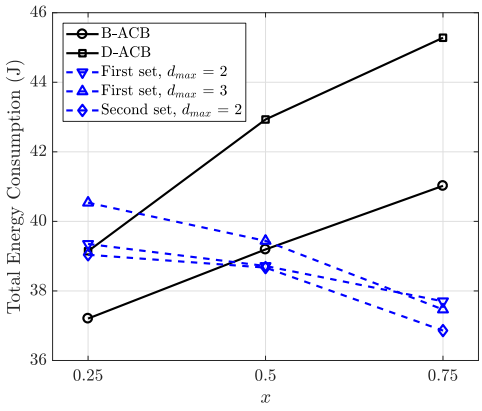

Fig. 11. Total energy consumption of all devices

Results obtained by scaling current PF set by a factor of 1.5 are shown in Table IV as bottom values in our entries. This second PF set places our proposal ahead of D-ACB in both $\mathbb{E}[D]$ of all classes and energy consumption as expected because increasing PF enhances classification effect without incurring additional capturing and hence, energy penalty. Note that our framework with fine-tuning still consumes about 5% more energy than B-ACB at $x = 0.25$. This scenario is however not of great interest in practice and the significant reduction in access delay of all classes easily justifies such cost. Moreover, PF can always be increased if a low consumption level is preferred over access delay.

V. CONCLUSION

In this paper, we have proposed a comprehensive DQ-based framework to effectively and practically tackle two major issues of cellular mMTC, namely massive access and device prioritization. To do so, the proposal made use of two core ideas. The first one was an estimation-aided DQ-based contention resolution protocol that tries to keep the number of devices per RAO at the optimal level in a best-effort attempt to maintain the RACH's efficiency. Then, the second idea of deferring RA procedure initialization of device classes by different amounts of time based on the queue's length was employed to realize dynamic prioritization. Simulation results under 3GPP reference setup verified the effectiveness of our framework's contention resolution capability and at the same time, showed that in the situation where MTDs of different priorities coexist, ours offered performance that is anywhere from comparable to significantly better than the ACB-based solutions in terms of delay and energy consumption. More importantly, being designed with practicality in mind, our framework fully complied to LTE specification and arose as a suitable access solution for cellular mMTC in 5G context.

APPENDIX A
DERIVATION OF $\omega(r)$

We define r as the *maximum expected* number of MTDs that can be provided with uplink grants in an RAO. Since the expected number of singleton preambles in an RAO with n_t transmitting MTDs is found as $\bar{S}$ in (1), it is obvious that $r = \bar{S}_{max}$. From (1), it is seen that $\bar{S}$ reaches its maximum when the number of transmitting devices equals to the number of preambles i.e., $n_t = K$. Thus, $r = \bar{S}_{max} = K(1-1/K)^{K-1}$. However, r is also upper capped by the maximum number of grants that can be sent out during a RAR window i.e., $W_{RAR} \times N_{RAR}$. Thus, we conclude that

$$r = \min\left\{K\left(1-\frac{1}{K}\right)^{K-1},\ N_{RAR}\times W_{RAR}\right\}.$$

To ensure the RACH's efficiency, it is necessary to keep the number of MTDs per RAO at a certain level $\omega(r)$ such that the corresponding number of singleton preambles $\bar{S}$ is approximately r. This desired $\omega(r)$ can thus be found as the $n_t \le K$ that minimizes the distance between $\bar{S}$ and r, i.e.,

$$\omega(r) = \arg\min_{n_t\in\mathbb{N}, n_t\le K} \left|\bar{S} - r\right|.$$

For the parameters in Table I i.e., $W_{RAR} \times N_{RAR} = 15$ and $K = 54$, we get $r = 15$ and $\omega(r) = 22$.

REFERENCES

[1] F. Ghavimi and H. 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, Secondquarter 2015.
[2] G. A. Akpakwu, B. J. Silva *et al.*, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.
[3] C. Anton-Haro and M. Dohler, *Machine-to-machine (M2M) Communications: Architecture, Performance and Applications*. Woodhead Publishing, 2015.
[4] J. Xu, J. Yao *et al.*, "Narrowband internet of things: Evolutions, technologies, and open issues," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1449–1462, June 2018.
[5] T. Taleb and A. Kunz, "Machine type communications in 3gpp networks: potential, challenges, and solutions," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 178–184, March 2012.
[6] M. Centenaro, L. Vangelista *et al.*, "Comparison of collision-free and contention-based radio access protocols for the internet of things," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3832–3846, Sept 2017.
[7] L. Tello-Oquendo, I. Leyva-Mayorga *et al.*, "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, April 2018.
[8] I. Leyva-Mayorga, L. Tello-Oquendo *et al.*, "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, Dec 2017.
[9] 3rd Generation Partnership Project (3GPP), *TR 37.868 V11.0.0, Study on RAN Improvements for Machine-type Communications*, Sep. 2011.
[10] A. Barki, A. Bouabdallah *et al.*, "M2m security: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1241–1254, Secondquarter 2016.
[11] X. Zhao, J. Zhai, and G. Fang, "An access priority level based random access scheme for qos guarantee in td-lte-a systems," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sept 2014, pp. 1–5.
[12] 3rd Generation Partnership Project (3GPP), *TS 36.321 V9.3.0, Medium Access Control (MAC) protocol specification*, Jun. 2010.
[13] S. Duan, V. Shah-Mansouri, and V. W. S. Wong, "Dynamic access class barring for m2m communications in lte networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2013, pp. 4747–4752.
[14] M. Tavana, V. Shah-Mansouri, and V. W. S. Wong, "Congestion control for bursty m2m traffic in lte networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2015, pp. 5815–5820.
[15] S. Duan, V. Shah-Mansouri *et al.*, "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.

[16] M. Bouzouita, Y. Hadjadj-Aoul *et al.*, "Applying nonlinear optimal control strategy for the access management of mtc devices," in *IEEE Annu. Consumer Commun. Networking Conf. (CCNC)*, Jan 2016, pp. 901–906.

[17] S. Lien, T. Liau *et al.*, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 27–32, January 2012.

[18] L. Tello-Oquendo, J. Vidal *et al.*, "Dynamic access class barring parameter tuning in lte-a networks with massive m2m traffic," in *Annu. Mediterranean Ad Hoc Netw. Workshop (Med-Hoc-Net)*, June 2018, pp. 1–8.

[19] 3rd Generation Partnership Project (3GPP), *TS 36.331 V10.5.0, Radio Resource Control (RRC) protocol specification*, Mar. 2012.

[20] T. M. Lin, C. H. Lee *et al.*, "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.

[21] T. Kim, K. S. Ko, and D. K. Sung, "Prioritized random access for machine-to-machine communications in ofdma based systems," in *2015 Proc. IEEE Int. Conf. Commun. (ICC)*, June 2015, pp. 2967–2972.

[22] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive multiple access based on superposition raptor codes for cellular m2m communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 307–319, Jan 2017.

[23] W. Xu and G. Campbell, "A near perfect stable random access protocol for a broadcast channel," in *Proc. IEEE Int. Conf. Commun.*, Jun 1992, pp. 370–374 vol.1.

[24] A. Laya, L. Alonso, and J. Alonso-Zarate, "Contention resolution queues for massive machine type communications in lte," in *Proc. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Aug 2015, pp. 2314–2318.

[25] A. T. H. Bui, C. T. Nguyen *et al.*, "An improved dq access protocol for cellular-based massive m2m communications," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Oct 2017, pp. 1–6.

[26] ——, "Design and performance analysis of a novel distributed queue access protocol for cellular-based massive m2m communications," *IEEE Access*, vol. 6, pp. 3008–3019, 2018.

[27] ——, "Free access distributed queue protocol for massive cellular-based m2m communications with bursty traffic," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Apr 2018.

[28] K. Lee and J. Wook Jang, "An efficient contention resolution scheme for massive iot devices in random access to lte-a networks," *IEEE Access*, vol. PP, pp. 1–1, 11 2018.

[29] 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 2014.

[30] M. Shahzad and A. X. Liu, "Fast and accurate estimation of rfid tags," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 241–254, Feb 2015.

[31] J. Vales-Alonso, V. Bueno-Delgado *et al.*, "Multiframe maximum-likelihood tag estimation for rfid anticollision protocols," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 487–496, Aug 2011.

[32] G. C. Madueño, N. K. Pratas *et al.*, "Massive m2m access with reliability guarantees in lte systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2015, pp. 2997–3002.

[33] H. Wu, C. Zhu *et al.*, "Fast adaptive s-aloha scheme for event-driven machine-to-machine communications," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sept 2012, pp. 1–5.

[34] H. Vogt, "Efficient object identification with passive rfid tags," in *Pervasive Computing*. Springer Berlin Heidelberg, 2002, pp. 98–113.

[35] L. Tello Oquendo, V. Pla *et al.*, "Efficient random access channel evaluation and load estimation in lte-a with massive mtc," *IEEE Tranns. Veh. Technol.*, vol. PP, pp. 1–1, 12 2018.

[36] 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, April 2015.

[37] U. Phuyal, A. T. Koc *et al.*, "Controlling access overload and signaling congestion in m2m networks," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Nov 2012, pp. 591–595.

[38] 3rd Generation Partnership Project (3GPP), *TS 36.211 V9.1.0, Evolved Universal Terrestrial Radio Access (E-UTRA) physical channels and modulation*, Mar. 2010.

[39] M. Gerasimenko, V. Petrov *et al.*, "Impact of machinetype communications on energy and delay performance of random access channel in lteadvanced," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 4, pp. 366–377.

[40] O. Arouk, A. Ksentini, and T. Taleb, "Group paging-based energy saving for massive mtc accesses in lte and beyond networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1086–1102, May 2016.

[41] A. R. Jensen, M. Lauridsen *et al.*, "Lte ue power consumption model: For system level energy and performance optimization," in *IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2012, pp. 1–5.

[42] O. Dementev, O. Galinina *et al.*, "Analyzing the overload of 3gpp lte system by diverse classes of connected-mode mtc devices," in *IEEE World Forum on Internet of Things (WF-IoT)*, March 2014, pp. 309–3012.