IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 2, 2017, accepted December 13, 2017, date of publication December 25, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2786678

Design and Performance Analysis of a Novel Distributed Queue Access Protocol for Cellular-Based Massive M2M Communications

ANH-TUAN H. BUI[1], (Senior Member, IEEE), CHUYEN T. NGUYEN[2], TRUONG C. THANG[1], (Member, IEEE), AND ANH T. PHAM[1], (Senior Member, IEEE)

[1]Computer Communications Laboratory, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[2]School of Electronic and Telecommunication, Hanoi University of Science and Technology, Hanoi 100000, Vietnam

Corresponding author: Anh T. Pham (pham@u-aizu.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant 15K00134 and in part by The University of Aizu's Competitive Fund P5.

ABSTRACT This paper proposes a novel access protocol based on the distributed queue (DQ) mechanism to effectively tackle the massive access issue in the cellular-based machine-to-machine (M2M) communications. To fully take the advantage of the DQ mechanism, we newly propose a method to avoid the DQ's inherent over-division problem by letting the base station first roughly probes the number of colliding devices in a random access opportunity. Based on the probing result, the base station then randomly divides these devices into a determined number of groups and "pushes" these groups to the end of a logical access queue. In addition, we develop an analytic model to accurately estimate the average access delay of the proposed protocol in the massive scenarios. Computer simulations are also performed to validate the correctness of the analytic model as well as the effectiveness of the proposed protocol in comparison with the LTE standard and conventional DQ access schemes.

INDEX TERMS Distributed queue, load estimation, LTE, massive M2M communications, random access protocols.

I. INTRODUCTION

Machine-to-Machine (M2M) communications, which supports a huge number of connected Machine-Type Devices (MTDs) operating in an autonomous manner, comes with not only invaluable chances for life-changing smart applications [1], [2], but also serious challenges to network foundation. To best accommodate the ubiquity of MTDs, the housing communications infrastructure must offer worldwide availability as well as enormous coverage. There are two main approaches currently being taken in an attempt to implement MTD over mobile cellular networks as depicted in Fig. 1 [2]. The first choice involves organizing MTD devices in a so-called M2M area networks using short-range wireless standards, such as Bluetooth, wifi. These M2M area networks are then connected to a mobile Base Station (BS) via M2M gateways equipped with Subscriber Identity Modules (SIMs). In this case, the gateways act as User Equipment (UE) from mobile network perspective. The second option is to directly incorporate MTDs into the existing mobile networks, such as existing LTE or the fifth generation (5G) networks [3], [4]. This approach not only eliminates the need for deployment of M2M-specific base stations, but also readily provides IP-native connectivity and thus, greatly facilitates the penetration of M2M into the market. This paper tackles the design and implementation issues of the second approach, which is also known as the cellular-based M2M communications.

One of the most challenging issues in the cellular-based M2M communications is the massive population of MTDs and their periodic access pattern [1], which may cause the Random Access Channel (RACH) of LTE to break down when thousands of devices try to access the network simultaneously. Recent studies have pointed out that in such scenario, the LTE RACH may experience heavy congestion, leading to very low access success probability [5]. Moreover, according to the 3GPP requirements, the cellular-based massive M2M is expected to have a capacity of 30K devices per cell, and for the 5G, a 10 times higher capacity, i.e. 300K devices per cell, should be envisioned [2]. Novel and more effective access protocols are therefore the key factor for the successful deployment of the M2M communications in the next-generation mobile networks.

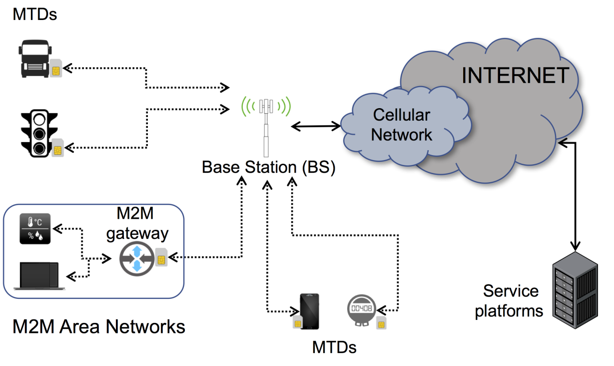


FIGURE 1. Cellular-based M2M communications.

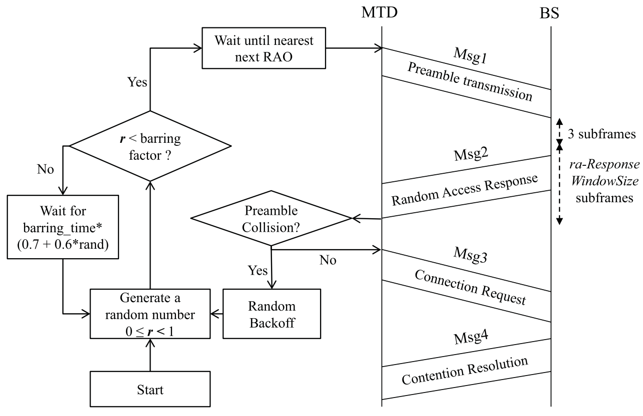


FIGURE 2. Standard LTE network access procedure with ACB scheme.

Over the past few years, there has been much effort focusing on improving the LTE standard protocols to deal with the M2M massive access issue [8]–[12]. As a matter of fact, the 3GPP has adopted an access protocol class called *Access Class Barring* (ACB) as the baseline solution to LTE RACH overload issue. ACB scheme categorizes devices into different "access classes" based on various criteria, and devices belonging to a class may perform access request in need with a certain probability called "barring factor" [6]. Simulation results provided in [7] with different MTD barring factors show that while the method does increase the access success probability under a relatively heavy load, the access delay of MTDs is severely degraded.

An alternative, promising solution is to employ the Distributed Queue (DQ) mechanism, a near-optimum contention resolution scheme [13]. The DQ-based protocols try to completely resolve an initial access collision using an *m*-ary splitting tree before attempting to handle subsequent ones and thus, effectively eliminates the uncertainty and instability seen in the LTE standard protocols [14]–[21]. Recent studies show that, compared to the standard ACB, the DQ-based protocol exhibits noticeable improvement under a relatively high load in comparison with the LTE standard ones. Nevertheless, the access delay is still prolonged in the massive access scenario (more than 1,500 and 2,500 simultaneous arrivals for the original DQ, namely, Contention Resolution Queue (CRQ), [16] and the improved one [20], [21], respectively) due to the over-division issue.

This has motivated us to propose a novel DQ-based random access mechanism to resolve the problem of M2M massive access, particularly when there may be more than ten thousands simultaneous arrivals. The key concept of our proposed protocol is to probe the number of colliding MTDs in each Random Access Opportunity (RAO) using a *partial estimation strategy* and randomly split these devices into a number of groups in a way such that the over-division issue in DQ-based protocols can be effectively avoided. To confirm the effectiveness of the proposal, we also develop an analytic model to estimate the average access delay of the proposed protocol over the Long-Term Evolution (LTE) networks with the massive access scenarios. The analytic results are also validated by computer simulations, and it is seen that, compared to the baseline ACB and CRQ schemes, the proposed protocol offers a significant delay reduction and reasonable average access delay in the low load and massive access scenarios, respectively.

The rest of our paper is organized as follows: In section II, we briefly introduce the standard random access procedure in LTE and the conventional DQ-based protocol. Our proposed protocol and its analytic model are presented in Section III and IV, respectively. Results and discussions are shown in section V and finally, section VI concludes the paper.

II. RANDOM ACCESS PROCEDURES

This section describes the overview of the standard LTE systems (the ACB protocol) and the DQ-based CRQ protocol. We also discuss their limitation and layout the background for the newly proposed one.

A. LTE RANDOM ACCESS CHANNEL (RACH)

Before fully examining the contention-based random access procedure, it is advisable to have a glimpse at the operation of the LTE RACH over which MTDs compete against each other to send their access requests.

The LTE RACH consists of a periodic sequence of time-frequency resources explicitly allocated for random access purpose known as Random Access Opportunities (RAOs). The temporal periodicity of these RAOs is encoded and broadcast to all MTDs via a parameter named *Physical RACH (PRACH) Configufation Index*. An example of this encoding is illustrated in Fig. 3. The 3GPP specifies 64 of such fixed mappings [22] where there can be a minimum of 1 RAO per 2 frames to a maximum of 1 RAO per 1 subframe.[1] It should be noted that the RAOs are implemented on the uplink physical channel and thus, the BS must carefully consider the compromises between the amount of allocated RAOs per frame and the amount of resources used for data transmission.

[1] A frame is composed of 10 subframes, each of duration 1ms.

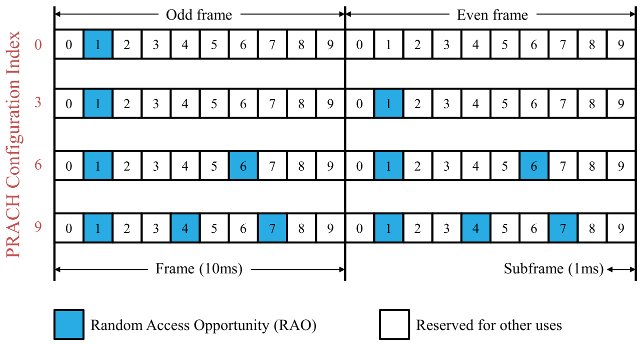


FIGURE 3. **PRACH configuration index.**

B. STANDARD LTE PROCEDURE

Now let us proceed to the details of the LTE contention-based Random Access (RA) procedure which, as depicted in Fig. 2, is a four-message handshake between MTDs and the BS. The role of each message is as follows.

Msg. 1, RA PREAMBLE : When an MTD needs to access the network, it first chooses one among a maximum of 64 orthogonal preamble sequences as its access request and sends that to the BS. This transmission takes place in the nearest future RAO, that is, over the RACH as discussed earlier. In this step, multiple MTDs of the same RAO may also choose the same preamble and cause a preamble collision that may or may not be discovered at the BS.

Msg. 2, RANDOM ACCESS RESPONSE (RAR): Three subframes after their preamble transmission RAO, the MTDs start looking for a response message known as the RAR for a time window of length *ra-ResponseWindowSize* subframes on the Physical Downlink Shared Channel (PDSCH). Note that the MTDs can pinpoint the location of their awaited RAR inside the PDSCH based on the location information of the RAO in which their preambles were sent. The RAR contains, for each successfully detected preamble, a 6-bit "identity" of that preamble, timing alignment instruction and initial uplink resource grant for Msg. 3. It may also include a Backoff Indicator (BI) to instruct the MTDs whose used preambles' identities are not found in the RAR to backoff for a random amount of time (based on BI) before attempting preamble retransmissions [23]. In the rare occasion where an MTD does not find any RAR during the designated time window at all, it restarts the RA procedure.

Msg. 3, CONNECTION REQUEST : Based on the instructions received in Msg. 2, MTDs transmit a connection request message containing the reason for access request and their identities. Here, undetected preamble collisions from Msg. 1 may result in more than one MTDs being granted the same resource to transmit Msg. 3 and cause a collision at the BS. In such case, colliding MTDs keep resending their Msg. 3 without success until a certain number of attempts is reached, then restart the RA procedure [23].

Msg. 4, CONTENTION RESOLUTION : This message is a response to MTDs whose Msg. 3 is received correctly by the BS and contains corresponding identities. At this point, the MTDs whose identities are found in Msg. 4 have successfully accessed the network and start scheduling their data transmissions, while the others restart the RA procedure. In the case where one Msg. 3 is decoded despite multiple Msg. 3 transmissions on the same time-frequency resource, only one MTD has its identity included in Msg. 4 and may progress [23].

Additionally, each MTD has a counter to keep track of the number of times it has sent a preamble. If this counter exceeds a threshold denoted by *preambleTransMax*, the MTD shall give up on the RA procedure and indicate a random access problem to upper layer. In such case, the MTD is said to be "blocked" from accessing the network [5].

When ACB is activated, an MTD first generates a random number in [0,1) and compare it with the MTD barring factor set by the BS. If the former is smaller, the UE may perform RA process; otherwise it has to wait for a random duration of $(0.7 + 0.6 * X)*Barring\_Time$, where X is another random number in [0,1), before retrying the test [6]. This mechanism helps spreading the load across the time domain to alleviate contention under high loads.

C. CONTENTION RESOLUTION QUEUE (CRQ) PROTOCOL

It is seen that in the LTE contention-based procedure, MTDs have to participate in an ALOHA-like contention process to transmit their preambles. At this point, the instability of ALOHA under high load has been proven [24]. Therefore, without carefully tuned barring parameters, an MTD following this mechanism may find itself continuously resending preambles without success due to consecutive collisions with a large number of backlogged MTDs before, eventually, blocked by the network.

CRQ is thus proposed to tackle this issue in the LTE supporting massive M2M [16]. The protocol organizes groups of devices that select the same preambles (i.e., colliding devices) in RAOs into a logical access "queue" based on the relative order of colliding preambles themselves. That is, when preamble collisions occur in an RAO, the group of MTDs selecting the lower-in-order colliding preamble is "pushed" to the end of the queue in advance of those who pick the higher-in-order one. Then in each RAO, given that the queue is non-empty, only the group of MTDs residing at the head of the queue may exit and re-perform the preamble transmission step. This splitting tree-like mechanism prevents MTDs involved in a collision from intervening in future retransmissions of MTDs involved in other collisions. As a result, the CRQ can eliminate the uncertainty factor to support an unlimited number of simultaneous arrivals.

An example of CRQ protocol with three available preambles is displayed in Fig. 4. In the first RAO, all devices collide and enter the queue. (u_1, u_7) occupy the first position since they used the lowest-in-order colliding preamble while (u_3, u_4) and (u_2, u_5, u_6) take up the second and third place, respectively. The two first groups then succeed in RAO 2 and 3 and exit the queue. The third group (u_2, u_5, u_6) retransmits in RAO 4, but only u_6 makes it to leave while the

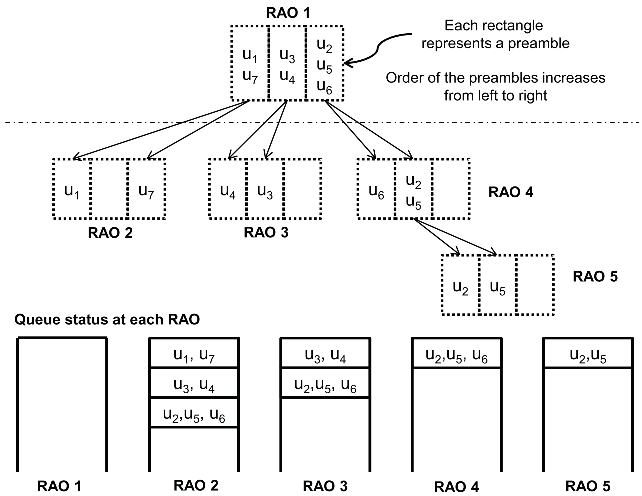


FIGURE 4. An example of CRQ protocol.

others collides and re-enters the queue. Finally, (u_2, u_5) leave the queue in RAO 5 where they succeed.

It is seen from Fig. 4 that each time a preamble collision occurs in an RAO, the CRQ, in essence, reserves a future RAO for the group of involved MTDs. This mechanism may cause excessive RAOs reservation when there are many preamble collisions occurring in an RAO, but each collision involves relatively few MTDs. That is, under such scenario, the CRQ inadvertently reserves a large number of RAOs (equal to the number of preamble collisions), but each RAO only serves an amount of involving MTDs way lower than its capability. This phenomenon, hereby referred to as *over-division*, is the main culprit behind the CRQ's performance inconsistency as proved in our earlier work [21]. To better visualize the impact of over-division, let us assume that there are 90 MTDs and 18 preambles to start with. Also let us assume that in the first RAO, each preamble is selected by 5 MTDs and ends up in a collision. Since there are 18 preamble collisions in total, the CRQ books 18 future RAOs (i.e., RAO 2 to RAO 19). Retransmitting in each of these RAOs, however, are only 5 MTDs, which are far less than the number of usable preambles i.e., 18. As a consequence, a huge number of unused preambles is generated and the delay is severely prolonged.

III. PROPOSED ACCESS PROTOCOL

As it is described in the last Section, that each of current approach has its own pros and cons in the considered context i.e., the massive M2M access. While the ACB scheme may work reasonably well in such case, it does so at the cost of a certain degree of instability which may result in excessive access delay of a small number of devices. Furthermore, improper setting of barring parameters may cause RACH overload when the number of devices exceed a certain mark. On the contrary, the CRQ protocol can theoretically support an unlimited number of devices [5]. Nevertheless, its over-division issue may cause excessive access delay for the whole population in massive multiple access scenario, as proved in our previous work [21]. This has motivated us to propose a novel design of DQ-based protocol that enjoys the advantages of both approaches to resolve the massive multiple access dilemma in an efficient manner.

The key concept of our protocol is to probe the number of colliding MTDs in each RAO using a *partial estimation strategy* and randomly split these devices into a number of groups in a way such that over-division can be avoided. In following sections, we describe the load estimation methods and the details of the proposed protocol.

A. LOAD ESTIMATION METHODS

Load estimation plays a crucial role in reducing random access delay, as it facilitates the design of optimal scheduling schemes. In fact, various estimation methods have been proposed to optimize the performance of random access protocols, albeit mainly in Radio Frequency IDentification (RFID) literature [25]. Recently, similar efforts have been made for cellular-based M2M with *fixed-location devices* (which is also our considered case in this paper). Recent work confirms that it is feasible to estimate the number of such devices in an RAO with a relatively good accuracy as long as they can maintain the same received power at the BS (which, practically, can be done via power control schemes) [26].

Particularly, the method in [26] divides the cell coverage into N_t regions, where devices in the same region have same delay to the BS. Then, each time transmissions from device(s) selecting an i-th preamble in the j-th spatial region are detected, the corresponding delayed version of the preamble i.e., $\mathcal{P}_{i,j}$ is subtracted from the received signal Y. This process cycles between all preambles until the power of Y drops below a threshold, and the number of devices selecting a particular preamble is found by counting the number of subtractions that have been made corresponding to that preamble.

The major issue with such iterative estimation method is that the needed number of iterations is roughly the same as the number of transmitting devices in the RAO. This leads to an excessive overhead delay, which may become prohibitive in massive access scenario with tens of thousands of simultaneous arrivals. We therefore propose an additional *partial estimation step* where we first probe the power level P of a randomly chosen i-th preamble by correlating it with the received signal Y as follows

$$P = \frac{1}{T}\sum_{j=1}^{N_t}\int_0^T Y(t)\mathcal{P}_{i,j}(t)dt = P_i + P_n, \tag{1}$$

where T is the duration of an OFDM symbol, P_i is the power from devices using the i-th preamble, and P_n is the noise power out of the integration-and-summation unit. To some extent, P can be used as a representative for the total number of transmitting devices. Thus, we may infer that if P is higher than a certain threshold γ, then the total number of transmitting devices is also likely to be more than a corresponding N_γ. In such case, we only perform this *partial estimation step*,

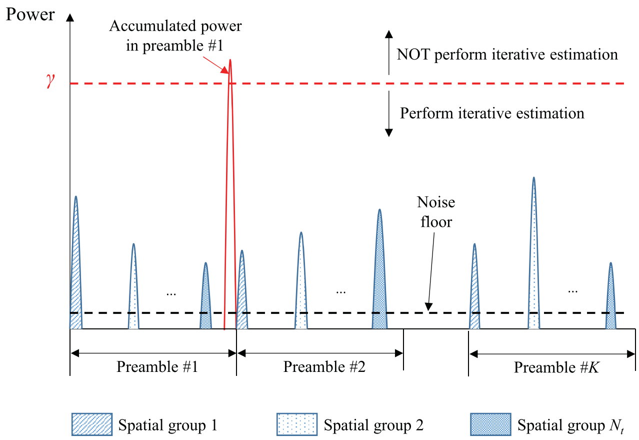


FIGURE 5. The proposed partial estimation strategy.

instead of the iterative estimation method as proposed in [26], in order to guarantee system response time. We will only perform the iterative estimation when the *partial estimation step* infers that the number of transmitting devices is less than N_γ. An example of this strategy is illustrated in Fig. 5 where the probed preamble is randomly chosen to be the first one.

It is noteworthy to mention that γ can be obtained via training for different values of N_γ and can be set at the BS. In this paper, we assume that γ is chosen corresponding to $N_\gamma = K^2$, where K is the number of available preambles. In other words, our key assumption is that whenever the number of transmitting MTDs in an RAO is higher than K^2, the BS will not necessarily perform the iterative estimation and thus, will not know the total number of accessing MTDs.

B. DQ-BASED ACCESS PROTOCOL WITH LOAD ESTIMATION

Now, we present our proposed DQ-based access protocol using load estimation methods to handle collisions under massive simultaneous access scenario. The key point of our protocol is that it controls the number of groups, G, which unsuccessful MTDs are divided into, so that the over-division issue can be effectively avoided. In particular, the BS first probes the number of transmitting MTDs i.e., N_r, in an RAO using our *partial estimation strategy* described earlier and select G as follows. Note that N_s here denotes the number of successful devices (or equivalently, the number of preambles selected by only one device) in an RAO.

- If $N_r > K^2$, then the number of devices are high enough so that the BS assumes $N_s = 0$. In this case, $G = K$. This does not cause over-division since the number of unsuccessful MTDs per group is still more than K.
- If $N_r \leq K^2$, perform iterative estimation to track the accurate value of N_r and N_s. Then $G = \left[(N_r - N_s) / K \right]$, where $[\cdot]$ denotes the *round* operator, so that the number of unsuccessful MTDs per group is K on average to prevent over-division. Furthermore, when $(N_r - N_s) < K$, the number of unsuccessful MTDs in this RAO is small enough that further division is always excessive. In such case, no more division is needed and thus, $G = 1$.

These G groups are then "pushed" to the end of a "distributed queue" to retry later. It is important to note that unsuccessful devices are not divided to groups based on their used preambles as in the CRQ, but rather in a random manner into a predetermined number of groups i.e., G.

As stated above, to realize the protocol, it is necessary to maintain a logical "distributed queue" wherein colliding devices are organized. This queue does not exist physically, but is facilitated via two parameters named DQ and pDQ. The former represents current "length" of the queue and is maintained by the BS while the latter is stored exclusively at each device to inform the device about its current "position" inside the queue. A device may transmit if its $pDQ = 0$ i.e., if it is at the queue's head. These parameters encode all relevant information on the queue and are updated after each RAO according to the following rules.

For DQ (at the BS):

- If $DQ > 0$, then DQ is decreased by 1 due to removal of the entry at the head of the queue.
- If preamble collisions occur, $DQ = DQ + G$ to reflect the addition of G groups of MTDs into the queue.

For pDQ (at individual MTD):

- If the device is waiting in the queue i.e., $pDQ > 0$, then $pDQ = pDQ - 1$ due to removal of the entry at the queue's head.
- If the device collides with others of the same preamble, it randomly select an integer g between [0, G - 1] to reflect its choice of group. Then its $pDQ = DQ - G + g$ to indicate that it, along with other devices of the g-th group, has re-entered the queue from the end. Note that G and DQ are included in corresponding Msg. 2.

Newly arrived MTDs must first observe the status of the ongoing process by listening to corresponding Msg. 2. If there are unsolved collisions ($DQ > 0$), they wait for DQ RAOs before re-checking Msg. 2 to acquire an updated value of DQ. This procedure is repeated until they find a Msg. 2 indicating that $DQ = 0$. In such case, the devices may invoke the access procedure in the very next RAO. Compared to the CRQ, these settings not only prevents newly arrived devices from interfering in the ongoing process, but also lower the energy consumption of such devices as they do not need to listen to Msg. 2 continuously.

Figure 6 presents an example of our protocol with $K = 6$ available preambles. Here, each rectangular represents an RAO while the upper and lower numbers in a rectangular represent the number of transmitting devices N_r and the number of unsuccessful devices i.e., $N_r - N_s$, in that RAO, respectively. In the very first RAO, $N_r = 90$ devices transmit simultaneously and cause collisions in all preambles. Via partial estimation step, the BS knows that $N_r > K^2$ in this RAO. Thus, it safely sets $G = K = 6$ to randomly break these 90 MTDs into 6 groups without triggering over-division. That is, the BS does so without worrying about

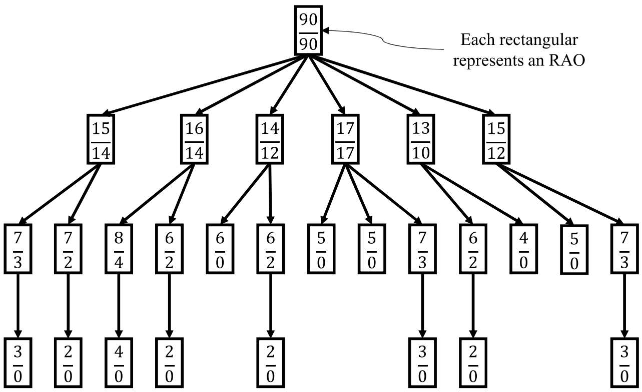


FIGURE 6. An example of the proposed protocol.

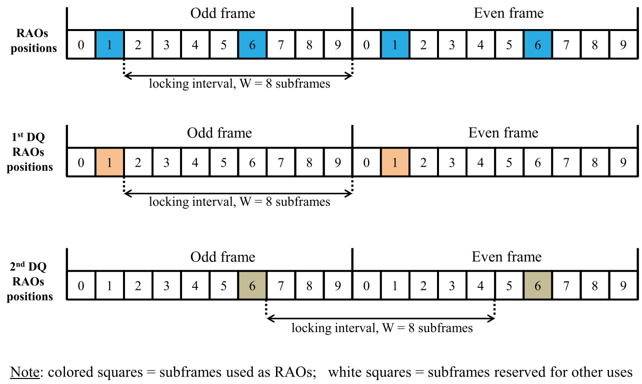


FIGURE 7. Example of multiple queues running in parallel.

the number of MTDs per group falling very short compared to K. The 6 groups then exit the queue in turn to re-perform preamble transmissions in the next 6 RAOs i.e., RAO 2 to RAO 7. Again, via partial estimation step, the BS is aware that the numbers of transmitting MTDs i.e., N_r, in each of these RAOs are lower than K^2. Thus, the BS first performs iterative estimation to obtain accurate N_r and N_s of each RAO. Then, to avoid over-division, the BS instructs the $N_r - N_s$ unsuccessful MTDs in each of these RAOs to split into $G = \lceil (N_r - N_s)/K \rceil$ groups so that the number of MTDs per group is approximately K. An example of this can be found by looking the second RAO where 14 unsuccessful MTDs are randomly divided into $G = \lceil 14/6 \rceil = 2$ groups with 7 MTDs (which are well around K) in each. All 13 groups created during RAO 2 to RAO 7 then consecutively exit the queue to retransmit in RAO 8 to RAO 20. As the number of MTDs in each of these 13 groups is approximately K, RAO 8 to RAO 20 achieve maximum efficiency (which occurs when $N_r \approx K$). The number of unsuccessful MTDs per RAO from that point also drops below K and thus, no further division is required and G is always 1 afterwards.

C. MULTIPLE DISTRIBUTED QUEUES AND PROPOSED ACCESS PROCEDURE

In the previous subsection, we have comprehensively described the operation of our proposed protocol which, however, only acts as a conceptual replacement to the underlying ALOHA-like framework of LTE. To be considered practical, the protocol must fully satisfy the timing requirements of the LTE's handshake procedure. Such task is actually non-trivial due to the reasons discussed in the following paragraph.

From Fig. 2, it is seen that after performing a preamble transmission, an MTD cannot take any further actions until the corresponding RAR-probing time window is over. This time window begins at exactly three subframes after the MTD's preamble transmission RAO and has a length of *ra-ResponseWindowSize* subframes, which results in a total locking interval of $W = 3 + \textit{ra-ResponseWindowSize}$ subframes. Consequently, any RAOs taking place during this interval is considered unusable for the transmitting MTD. This barely affects the ACB scheme, as RAOs unusable for a transmitting MTD may still be utilized by others who just resume from their previous backoff or barring wait. On the contrary, DQ-based protocols require that all MTDs (that is, both transmitting and waiting ones) do not adjust their counters until transmitting MTDs have received their RAR i.e., until the locking interval W of transmitting MTDs has elapsed, to ensure that the queue operates in a synchronous manner. Thus, in effect, a single distributed queue can only use RAOs that are at least W subframes apart from each other. This leads to under-utilization when the RAOs are spaced at less than or equal to W subframes.

To overcome this issue, we propose to facilitate more than one distributed queues running in parallel so that RAOs unusable by a queue will be exploited by the others. Our idea is best described via the example provided in Fig. 7 where *ra-ResponseWindowSize* is set to 5 subframes [27], and the RAOs are located at subframes number 1 and 6 of any frames (which is equivalent to setting *PRACH Configuration Index* to 6). If there is only one queue, it is easy to see that to satisfy the RAOs spacing condition i.e., consecutive usable RAOs of a queue must be spaced at least W subframes apart, the queue can only use either the pair of subframes number 1 or 6, which leads to only 50% RAOs utilization. To achieve full RAOs utilization while still secure the RAOs spacing constraint, we propose the parallel use of two separate queues where the first occupies the pair of subframes number 1 and the second takes the pair of subframes number 6 as depicted in the middle and bottom of Fig. 7, respectively. This concept extends naturally to other PRACH configurations. Also, we assume that the number of parallel queues as well as the allocated positions of their RAOs inside a frame are broadcast to all MTDs by the BS as part of system information. Upon entering the network, an MTD will randomly select a queue to associate with.

The access procedure, modified to accommodate our proposed protocol, can finally be visualized as in Fig. 8.

IV. ANALYTIC MODEL & PERFORMANCE ANALYSIS

In this section, we propose an analytic model to study the access delay of our proposed protocol. The operation of the

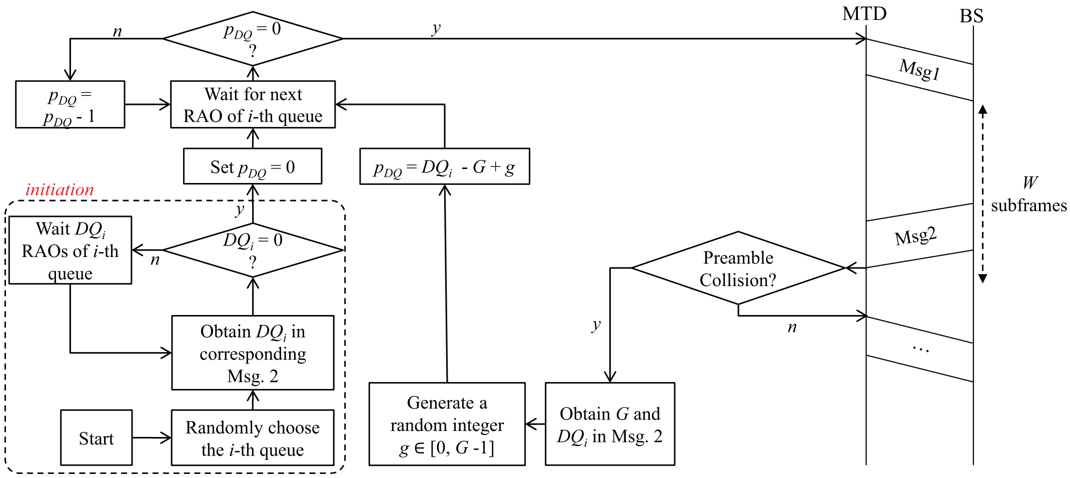


FIGURE 8. Access procedure representation of the proposal.

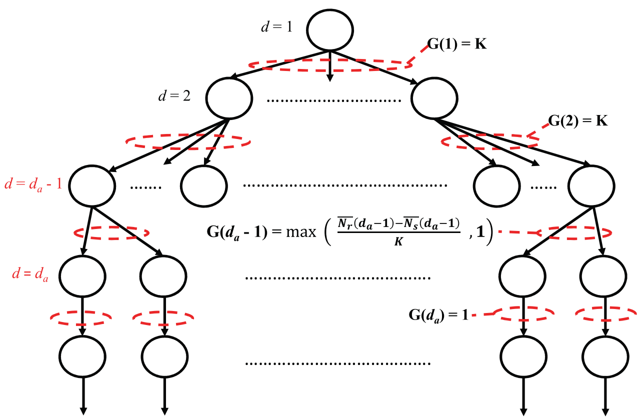


FIGURE 9. Analytic delay model of the proposed protocol.

proposed protocol can be visualized as an m-ary splitting tree in Fig. 9, where each node corresponds to an RAO, and the number of children of a node corresponds to the number of groups which unsuccessful devices in that node are divided into. Here, we perform approximation by assuming that every nodes of the same depth d has the same *average* number of children denoted by $G(d)$, and the number of devices in each child node is also the same. $G(d)$ can be written as

$$G(d) = \begin{cases} \max\left(\dfrac{\overline{N_r}(d) - \overline{N_s}(d)}{K}, 1\right), & \overline{N_r}(d) \leq K^2 \\ K, & \overline{N_r}(d) > K^2 \end{cases} \tag{2}$$

where $\overline{N_r}(d)$ and $\overline{N_s}(d)$ are the average number of transmitting devices and successful devices in a node of depth d, respectively. $\overline{N_s}(d)$ is related to $\overline{N_r}(d)$ as $\overline{N_s}(d) = \overline{N_r}(d)P_s(d)$, where $P_s(d)$ is the probability of success of a device given that it is at a node of depth d. $P_s(d)$ may be interpreted as the probability that no other devices in the node selects the same preamble as this device and thus, is expressed as

$$P_s(d) = \left(1 - \frac{1}{K}\right)^{\overline{N_r}(d)-1}. \tag{3}$$

As seen from the Fig. 9, if an MTD succeeds at a depth d, it will have waited for a number of RAOs equal the total number of nodes from all previous depths plus, on average, half of the number of nodes of d (as it may belong to any node of d with equal probabilities). For convenience, we denote the number of nodes of a depth d by $L_n(d)$ and the average number of elapsed RAOs for a device succeeding at d by $L(d)$. The relationship between the two is expressed as

$$L(d) = \sum_{d'=1}^{d-1} L_n(d') + \frac{L_n(d)}{2}. \tag{4}$$

In order to find $L_n(d)$, we must know $G(d')$ for all $d' < d$ as they directly affect the number of nodes at each depth upto d. It is evident that $G(d') = K$ for all depths d' at which $\overline{N_r}(d') > K^2$. On the other hand, at a particular depth $(d_a - 1)$, the number of devices per node start dropping below K^2 and thus,

$$G(d_a - 1) = \max\left(\frac{(1 - P_s(d_a - 1))\,\overline{N_r}(d_a - 1)}{K}, 1\right). \tag{5}$$

This makes the average number of devices per node at the very next depth $\overline{N_r}(d_a)$ approximately K and thus, no more division is needed afterwards i.e., $G(d') = 1$ for all $d' \geq d_a$. Having found all $G(d)$ based on d_a, we then proceed to calculate $L_n(d)$ as follows. Note that it is trivial that $L_n(1) = 1$.

- For $2 \leq d \leq (d_a - 1)$:

$$L_n(d) = \prod_{d'=1}^{d-1} G(d') = K^{d-1}. \tag{6}$$

TABLE 1. Simulation parameters.

Parameter	Simulated Values
Number of available preambles K (associated barring factor)	56 (0.8) , 36 (0.6) , 18 (0.4)
Barring time T_{bar}	1s , 1.5s , 2s
Backoff indicator	480ms
ra-ResponseWindowSize (resultant locking interval W)	5 subframes (8 subframes)
Spacing between consecutive RAOs of a queue	9 subframes
PRACH Configuration Index (resultant number of queues Q)	3 (1) , 6 (2) , 9 (3)
Maximum allowed number of preamble retransmissions *preambleTransMax*	10

- For $d = d_a$, $L_n(d)$ is rewritten as

$$L_n(d_a) = G(d_a - 1) \prod_{d'=1}^{d_a-2} G(d') = G(d_a - 1)K^{d_a-2}. \quad (7)$$

- For $d > d_a$, there is no further splitting and thus,

$$L_n(d) = L_n(d_a) = G(d_a - 1)K^{d_a-2}. \quad (8)$$

It is clear that d_a must be derived for specific calculation of $L(d)$. To find d_a, we first observe that $P_s(d) \approx 0$ and $G(d) = K$ for all $d < d_a - 1$. Then, $\overline{N_r}(d_a - 1)$ can be found as

$$\overline{N_r}(d_a - 1) = \frac{N}{K^{d_a-2}}. \quad (9)$$

It is also reminded that at $d_a - 1$, the number of devices per node starts dropping below K^2 i.e., $\overline{N_r}(d_a - 1) < K^2$. Thus, $N < K^{d_a}$ and since d_a is the smallest integer that satisfies this condition, we can conclude that $d_a = \lceil \frac{\log(N)}{\log(K)} \rceil$. It is important to note that $G(d_a - 1)$ is consequently derived by substituting $\overline{N_r}(d_a - 1)$ in (9) into (5) and (3). This completes the calculation for $L(d)$.

Now that $L(d)$ is found, the remaining task is to derive the probability that an MTD succeeds at d denoted by $Pr(d)$. This is interpreted as the probability that the device fails at all depths prior to d and then succeeds at d, which is then expressed as

$$Pr(d) = P_s(d) \prod_{d'=1}^{d-1} \left(1 - P_s(d')\right). \quad (10)$$

As seen from (3), $\overline{N_r}(d)$ is required to calculate $P_s(d)$ for any d. In general, $\overline{N_r}(d)$ can be written as

$$\overline{N_r}(d) = \frac{(1 - P_s(d-1))\,\overline{N_r}(d-1)}{G(d-1)}. \quad (11)$$

Therefore, $\overline{N_r}(d)$ is a function of $\overline{N_r}(d-1)$. This establishes a recursive solution to find $\overline{N_r}(d)$ for any d, where the initial condition is $\overline{N_r}(1) = N_q$ where N_q denotes the total number of simultaneous arrivals at the MTD's choosen queue. Thus, $Pr(d)$ for any d is consequently derived. Finally, the average number of RAOs (or correspondingly, the average access delay) that an MTD has to wait, denoted by $D(N_q)$, is found as

$$D(N_q) = \sum_{d=1}^{\infty} L(d)Pr(d). \quad (12)$$

Note that $D(N_q)$ is measured in the unit of RAOs of the MTD's chosen queue. That is, these RAOs must be spaced at least W subframes apart of each other. To provide a more practical result, we fix the value of W to 8 subframes as in Fig. 7 and the spacing between consecutive RAOs of a queue to 9 subframes (which is well longer than W) so that there is one RAO per one frame i.e., one RAO every 10ms, for every queue. The average delay that an MTD has to wait until its successful preamble transmission, measured in the unit of seconds, is then easily found as $0.01 * D(N_q)$

Finally, let us assume that there are N simultaneous arrivals to the system and Q queue(s) running in parallel. Since the MTDs randomly choose one among Q queue(s) to perform access, each queue serves on average $N_q = N/Q$ simultaneous arrivals. Furthermore, since the queues run in parallel, the average delay of an MTD of any queue is also the average delay of all MTDs. That is, the average delay that an MTD has to wait until its successful preamble transmission given N and Q, measured in the unit of seconds, is written as $0.01 * D(N/Q)$.

V. RESULTS & DISCUSSIONS

In this section, computer simulations using basic MATLAB programming are performed to validate our analytic model for delay analysis, as well as to demonstrate the effectiveness of the proposed protocol in comparison with standard ACB scheme and the CRQ protocol. Configuration parameters for the RA procedure are summarized in Table 1 and are assumed to be known by all MTDs. As for the traffic model, we employ *batch arrivals* with the following assumptions (which are reasonable for M2M traffic):

- N MTDs try to access the network simultaneously every T_p seconds, where T_p is the period of the batches.
- There is no new arrivals between two consecutive batches.
- T_p is long enough that all collisions from previous batch has been resolved by the time a new batch arrive.

The protocols of interest are then assessed using access delay as the main performance metric. Note that we only observe the random access delay of an MTD until when it successfully sends its preamble, as following Msg. 2, 3 and 4 transmissions are not random but scheduled by the BS. The access delay of the MTD, in the unit of seconds, is then found as 0.01 × the number of RAOs of the chosen queue that has elapsed until the MTD's successful preamble transmission.

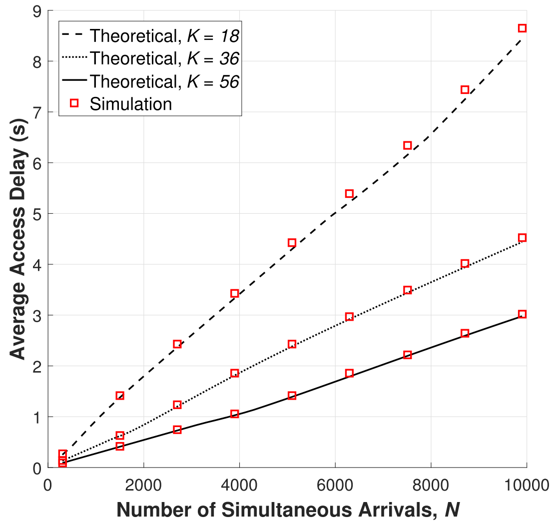


FIGURE 10. **Analytical vs. simulation result of average access delay of the proposed protocol, given *PRACH Conf. Index* = 3.**

It is also worth mentioning that the average access delay is calculated over *non-blocked* MTDs only. This means that the delay of blocked MTDs i.e., MTDs who have exceeded the maximum allowed number of preamble retransmissions, is not considered.

We first validate the analytic delay model by the computer simulation in Fig. 10 that shows both the analytic and obtained simulation results of the average access delay of the proposed protocol with the number of arrivals, N, of up to 10,000 and *PRACH Configuration Index* set to 3. It is seen that there is a good agreement between the two in all cases, which validates the correctness of the analytic model. Using this model, ones can quickly estimate the average access delay of MTDs with a specific system configuration.

Next, we confirm the effectiveness of the proposed protocol, in terms of average access delay, in comparison with the LTE standard scheme (ACB) and the conventional DQ-based CRQ protocol under the settings of *PRACH Configuration Index* $= 3$ and $T_{bar} = 2$ seconds. Initially, in Fig. 11a we compare all three protocols in the load range of up to 2,100 simultaneous arrivals. The result clearly shows that our proposed protocol offers a significant improvement compared to both ACB and CRQ at any level of load. As an example, at the number of arrivals $N = 1500$, the advantages of ours over the CRQ are seen to be 37%, 65% and 54% when there are 18, 36 and 56 preambles, respectively. Compared to the ACB under the same conditions, these numbers respectively become 71%, 75% and 68%. The result also confirms the fact, as also can be seen in [16], that the DQ-based protocol (CRQ) outperforms the ACB in this range of load. Nevertheless, as the load gets higher, the CRQ suffers from higher access delay than that of the ACB due to excessive reservation issue which becomes apparent. It is shown that, at $N = 1,900$, the ACB starts outperforming the CRQ. Thus, for better clarity, we omit the CRQ and use the ACB as reference in massive simultaneous arrivals scenario.

Delay performance of our protocol and the ACB scheme in the high load region is found in Fig. 11b where we assess up to around 15,000 simultaneous arrivals. It is seen that the proposal still shows a remarkable gain in the range of $1,900 \leq N \leq 5,000$. However, as the system enters the massive multiple access region of $N > 5000$, the performance gap between the protocols becomes gradually narrower. This trend continues until the ACB scheme reaches a breakpoint at which its average delay is dangerously close to ours. Nevertheless, the average access delay of ACB starts to increase rapidly beyond these marks, which suggests that its RACH has been overloaded. On the contrary, our protocol still keeps a stable delay performance to re-widen the advantage gap. As an example, at $N = 14,700$ and $K = 56$, this gap has already been re-expanded to 16%.

We also note that the breakpoints of ACB for different K are different, where $K = 18$, 36 and 56 show breakpoints at $N = 13{,}100$, 11,500 and 9,900 respectively. This may seem odd at first since the cases with higher K reach their breakpoints sooner. However, it is important to keep in mind that the barring factor is much smaller when K is low, which greatly disperses access attempts on the time domain. In such case, the RACH can accommodate more MTDs at the cost of significantly higher access delay.

Figure 12 is then plotted in an attempt to investigate the access delay of the system under different LTE settings at the peak load of the considered massive access region, i.e. $N = 14,700$. Here, the immediately recognized feature is that even with different LTE configurations, our protocol manages to surpass the ACB scheme in most cases. Furthermore, observed access delay of the latter appears to vary in an arbitrary manner. To better understand such dynamic, it is recalled that the access delay is averaged over *non-blocked* MTDs only. Thus, the access delay should be viewed in conjunction with the blocking probability, defined as the ratio of the number of blocked MTDs to the total number of MTDs N and used as an indicator for the severity of access congestion, for a complete picture.

The blocking probabilities corresponding to Fig. 12 is displayed in Table 2.[2] For the sake of discussion, we henceforth consider the system (using ACB) as in *non-overloaded* state if the blocking probability is approximately 0%, and in *overloaded* state otherwise. It is consequently seen that whenever the system is non-overloaded e.g., in most of the cases associated with *PRACH Configuration Index* $= 6$ or 9, ACB scheme at lower T_{bar} performs very similar to the proposal. As T_{bar} grows, however, access delay of the former is prolonged, causing it to fall short against our protocol. This is because being in non-overloaded state implies that the system with ACB is capable of handling the offered load, and increasing T_{bar} thus unnecessarily defers MTDs' (re)transmissions without significantly improving the already infinitesimal blocking probability.

[2]Our protocol does not have blocking under the simulation conditions in Table 1 and thus, is dropped from this table

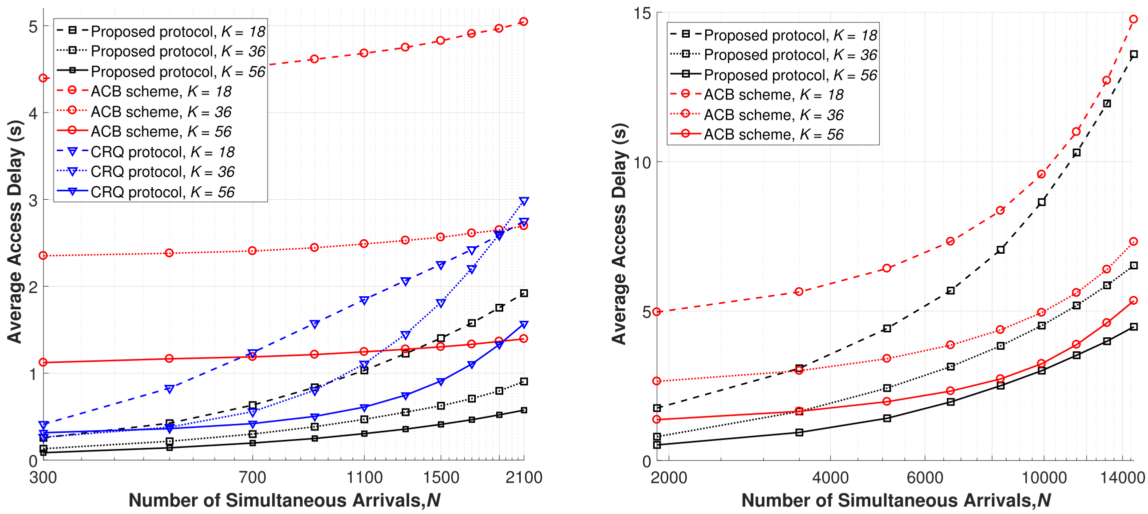


FIGURE 11. Average access delay comparison of the protocols, given *PRACH Conf. Index* = 3 and T_{bar} = 2s. (a) Lower access region. (b) Massive access region.

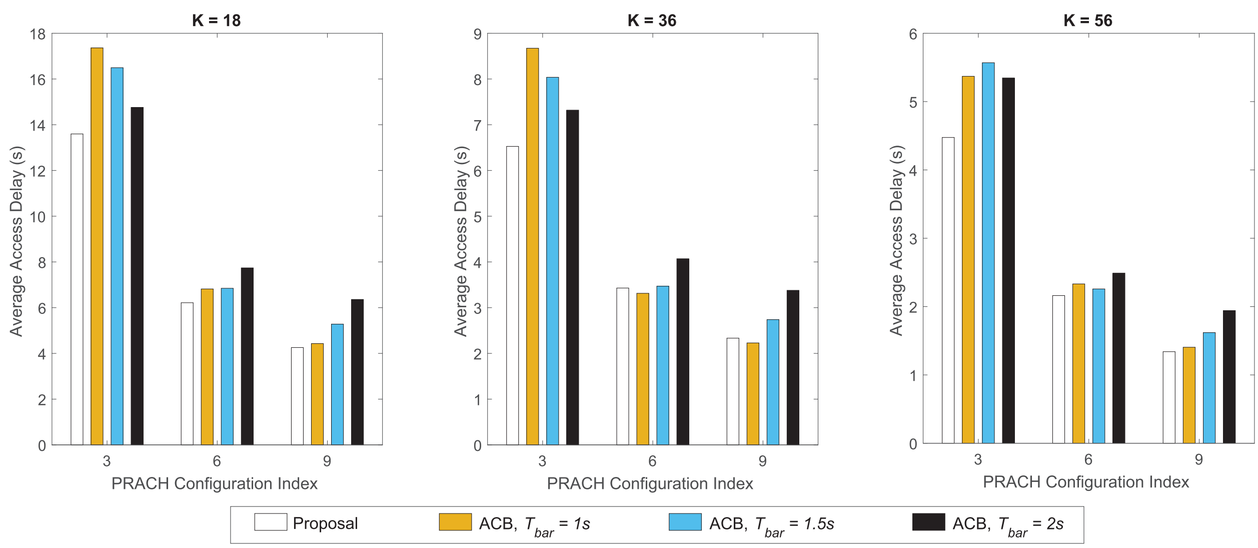


FIGURE 12. Average access delay of the proposal and ACB under peak load i.e., $N = 14700$.

TABLE 2. Blocking probability of ACB schem under peak load i.e., $N = 14700$.

(a) K = 18

T_{bar} \ Conf. Index	3	6	9
1 second	46.67%	0.34%	0%
1.5 seconds	10.49%	0.01%	0%
2 seconds	1.07%	0%	0%

(b) K = 36

T_{bar} \ Conf. Index	3	6	9
1 second	41.98%	0.26%	0%
1.5 seconds	10.22%	0.01%	0%
2 seconds	1.39%	0%	0%

(c) K = 56

T_{bar} \ Conf. Index	3	6	9
1 second	56.47%	2.15%	0.02%
1.5 seconds	34.55%	0.3%	0%
2 seconds	16.38%	0.05%	0%

When the system is overloaded, e.g. *PRACH Configuration Index* is 3, our proposed protocol outpaces the ACB scheme by a remarkable margin for all tested T_{bar}. Moreover, we can safely assure that although access delay of the latter seems to gradually drops as the barring time increases, it cannot drop below ours even if T_{bar} keeps getting prolonged. This claim is backed up by the fact that at T_{bar} settings where the observable performance gap is smallest i.e., at $T_{bar} = 2$s, the blocking probabilities of the ACB is already very close to 0%. That is, further increment in T_{bar} quickly puts the system back to the non-overload state in which the delay is bounded to rise as T_{bar} gets higher.

Finally, one may conclude from Fig. 12 that in the case of very high load, i.e. massive arrivals from MTDs, our protocol shows a performance level that is anywhere from comparable with to significantly better than the ACB. What we want to emphasize here is that the ACB scheme requires almost perfect coordination between *all LTE parameters* in accordance with a specific load to achieve the performance level similar to what ours has to offer.

VI. CONCLUSIONS & FUTURE WORK

In this paper, the foreseen massive multiple access issue in the context of cellular-based M2M communications had been throughout studied. It was seen that in dense scenario where hundreds of thousands of MTDs are expected to be housed by a single base station, simultaneous access attempts from a measly 1% of the population could still impose significant stress on the current-gen LTE networks' random access channel. Several existing countermeasures, including both the 3GPP's ACB scheme and the CRQ protocol, as well as their pros and cons, were also investigated. More importantly, we then introduced a novel DQ-based random access protocol coupled with an estimation strategy to address the issue in a more efficient manner. By randomly dividing unsuccessful MTDs in an RAO into a number of groups based on the estimation strategy's output and pushing these groups to the end of a logical access queue to retry later, our protocol was able to avoid both the over-division problem of the CRQ and the uncertainty factor of the ACB. The use of multiple access queues running in parallel was also suggested to fully exploit the closely-spaced RAOs. Finally, we constructed a delay model for the proposal and performed computer simulations to validate both the analytic model and the effectiveness of the proposed protocol over the ACB and the CRQ.

The simulation results showed that the CRQ initially outperformed the ACB scheme, but quickly lost its favor when the load increased due to over-division. Our protocol, however, outpaced both the ACB and CRQ in a wide range of load up to around 15,000 simultaneous access attempts, given a certain LTE configuration. Under closer inspection at peak load, it was seen that even with different LTE configurations, our protocol exhibited a performance level ranging from comparable with to remarkably better than the ACB. Thus, the proposed protocol emerged as one highly feasible solution for massive M2M access scenario, when the CRQ suffered from the severe impact of over-division and the ACB started getting overloaded.

Finally, it is worth noting that besides the simultaneous access pattern considered in this paper, there exists other approaches to model MTDs' access patterns as well. In fact, the 3GPP has readily provided two references, quoted as the uniform and Beta traffic models, to depict the access patterns of uncorrelated and highly correlated MTDs, respectively. These models assume that the MTDs wake up to perform random access following certain temporal distributions over a fixed period, which is significantly different from the studied simultaneous case where the MTDs access at the same time instance. Therefore, in our future works, we are interested in seeing how our protocol fares compared to the ACB and other novel access schemes under the 3GPP's reference setups.

REFERENCES

[1] D. Boswarthick *et al.*, *M2M Communications: A Systems Approach*. Hoboken, NJ, USA: Wiley, 2012.

[2] M. Dohler *et al.*, *5G Mobile and Wireless Communications Technology*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[3] *Study on Facilitating Machine to Machine Communication in 3GPP Systems*, document 3GPP TR 22.868 V8.0.0, Mar. 2007.

[4] *Machine to Machine Communications (M2M); Functional Architecture*, document ETSI TS 102.690 V2.1.1, Oct. 2013.

[5] 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, 1st Quart., 2013.

[6] *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)*, document 3GPP TS 36.331 V10.5.0, Mar. 2012.

[7] *MTC Simulation Results With Specific Solutions*, document 3GPP TSG RAN WG2 # 71 R2-104662, Aug. 2010.

[8] P. Si, J. Yang, S. Chen, and H. Xi, "Adaptive massive access management for QoS guarantees in M2M communications," *IEEE Trans. Veh. Technol.*, vol. 64, no. 7, pp. 3152–3166, Jul. 2015.

[9] A. Ilori, Z. Tang, J. He, K. Blow, and H.-H. Chen, "A random channel access scheme for massive machine devices in LTE cellular networks," in *Proc. IEEE ICC*, London, U.K., Jun. 2015, pp. 2985–2990.

[10] 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.

[11] G.-Y. Lin, S.-R. Chang, and H.-Y. Wei, "Estimation and adaptation for bursty LTE random access," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2560–2577, Apr. 2016.

[12] J. Mišić, V. B. Mišić, and N. Khan, "Sharing it my way: Efficient M2M access in LTE/LTE-A networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 696–709, Jan. 2017.

[13] W. Xu and G. Campbell, "A near perfect stable random access protocol for a broadcast channel," in *Proc. IEEE ICC*, Jun. 1992, pp. 370–374.

[14] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiró, and L. Alonso, "Energy performance of distributed queuing access in machine-to-machine networks with idle-to-saturation transitions," in *Proc. IEEE GLOBECOM Wkshps*, Atlanta, GA, USA, Dec. 2013, pp. 867–872.

[15] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiró, and L. Alonso, "Energy analysis of a contention tree-based access protocol for machine-to-machine networks with idle-to-saturation traffic transitions," in *Proc. IEEE ICC*, Sydney, NSW, Australia, Jun. 2014, pp. 1094–1099.

[16] A. Laya, L. Alonso, and J. Alonso-Zarate, "Contention resolution queues for massive machine type communications in LTE," in *Proc. IEEE PIMRC Workshop M2M Commun., Challenges, Solutions Appl.*, Sep. 2015, pp. 2314–2318.

[17] A. Laya, L. Alonso, and J. Alonso-Zarate, "Efficient contention resolution in highly dense LTE networks for machine type communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–7.

[18] F. Vázquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Energy harvesting-aware distributed queuing access for wireless machine-to-machine networks," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–7.

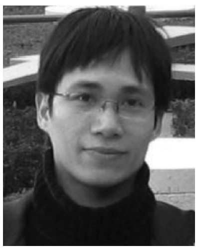
[19] F. M. Awuor and C.-Y. Wang, "Massive machine type communication in cellular system: A distributed queue approach," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–7.

[20] A.-T. H. Bui and A. T. Pham, "Efficient access protocol for massive M2M communications in LTE," in *Proc. IEICE General Conf.*, Mar. 2017, pp. BS-1–BS-34.

[21] A.-T. H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "An improved DQ access protocol for cellular-based massive M2M communications," in *Proc. IEEE/CIC ICC China*, Qingdao, China, Oct. 2017.

[22] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*, document 3GPP TS 36.211 V10.4.0, Dec. 2011.
[23] *Evolved Universal Terrestrial Radio Access (EUTRA); Medium Access Control (MAC)*, document 3GPP TS 36.321 V9.3.0, Jun. 2010.
[24] S. Ghez, S. Verdú, and S. C. Schwartz, "Stability properties of slotted Aloha with multipacket reception capability," *IEEE Trans. Autom. Control*, vol. 33, no. 7, pp. 640–649, Jul. 1988.
[25] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, ALOHA!" *IEEE Access*, vol. 4, pp. 2029–2044, Apr. 2016.
[26] 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.
[27] C. Anton-Haro and M. Dohler, *Machine-to-Machine (M2M) Communications: Architecture, Performance and Applications*. Cambridge, U.K.: Woodhead Publishing, 2015.

ANH-TUAN H. BUI (SM'17) received the B.E. degree in communications engineering from the Hanoi University of Science and Technology, Vietnam, in 2016. He is currently pursuing the M.E. degree with The University of Aizu, Japan. He received the Japanese Government Scholarship (Monbukagakusho). His research interests are in the field of protocol designs, system modeling, and performance evaluation for next-generation (5G) wireless networks.



CHUYEN T. NGUYEN received the M.S. degree in communications engineering from National Tsing-Hua University, Taiwan, in 2008, and the Ph.D. degree in informatics from Kyoto University, Japan, in 2013. In 2014, he joined The University of Aizu, Japan, as a Visiting Researcher. He is currently an Assistant Professor with the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam. His research interests include statistics and optimization algorithms and their applications in wireless communication systems. He received the Fellow Award from the Hitachi Global Foundation in 2016.

TRUONG C. THANG received the B.E. degree from the Hanoi University of Science and Technology, Vietnam, in 1997, and the Ph.D. degree from KAIST, South Korea, in 2006. From 1997 to 2000, he was a Network Engineer with Vietnam Post and Telecom. From 2007 to 2011, he was a Research Staff Member with the Electronics and Telecommunications Research Institute, South Korea. He was also an active member of South Korean and Japanese delegations to standard meetings of ISO/IEC and ITU-T from 2002 to 2014. Since 2011, he has been an Associate Professor with The University of Aizu, Japan. His research interests include multimedia networking, image/video processing, content adaptation, IPTV, and MPEG/ITU standards.

ANH T. PHAM (M'00–SM'11) received the B.E. and M.E. degrees in electronics engineering from the Hanoi University of Technology, Vietnam, in 1997 and 2000, respectively, and the Ph.D. degree in information and mathematical sciences from Saitama University, Japan, in 2005. From 1998 to 2002, he was with NTT Corporation, Vietnam. Since 2005, he has been a Faculty Member with The University of Aizu, where he is currently a Professor and the Head of the Computer Communications Laboratory, Division of Computer Engineering. His research interests are in the broad areas of communication theory and networking with a particular emphasis on modeling, design, and performance evaluation of wired/wireless communication systems and networks. He has authored or co-authored over 160 peer-reviewed papers on these topics. He is a member of IEICE and OSA.