D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks

Suyang Duan, Vahid Shah-Mansouri, *Member, IEEE*,
Zehua Wang, *Student Member, IEEE*, and Vincent Wong, *Fellow, IEEE*

***Abstract*—When incorporating machine-to-machine (M2M) communications into the Third Generation Partnership Project (3GPP) Long Term Evolution (LTE) networks, one of the challenges is the traffic overload since many machine-type communication (MTC) devices activated in a short period of time may require access to an evolved node B (eNodeB) simultaneously. One approach to tackle this problem is using access class barring (ACB) mechanism with an ACB factor to defer some activated MTC devices transmitting their access requests. In this paper, we first present an analytical model to determine the expected total service time, *i.e.*, the time used by all MTC devices to successfully access to the eNodeB. In the ideal case that the eNodeB is aware of the number of backlogged MTC devices, we determine the optimal value of the ACB factor to reduce traffic overload. To better utilize the random access resources shared between human users and MTC devices in LTE networks, we propose to dynamically allocate a number of random access preambles for MTC devices. We further propose two dynamic access class barring (D-ACB) algorithms for fixed and dynamic preamble allocation schemes to determine the ACB factors without *a priori* knowledge of the system backlog. Simulation results show that the proposed D-ACB algorithms achieve almost the same performance as the optimal performance obtained in the ideal case. The proposed D-ACB for dynamic preamble allocation algorithm can reduce both the total time to serve all MTC devices and the average number of random access opportunities required by each user equipment.**

***Index Terms*—Machine to machine (M2M) communications, random access control, LTE networks**

I. Introduction

THE machine-to-machine (M2M) communication network is a communication network that includes a large number of machine-type communication (MTC) devices that can communicate with each other or remote servers without human interventions to accomplish specific tasks. M2M communications enable the implementation of the Internet of things, in which ubiquitous connections can be established either on demand or in a periodic manner [1]. It is expected that there will be 12.5 billion MTC devices (excluding smartphones and tablets) by 2020 [2].

M2M communications have a wide range of applications, including vital sign monitoring in health care systems, monitoring of the oil pipelines, on-demand charging transactions in e-commerce, fleet management, and communications of smart meters in smart grid [1]. Although communications between MTC devices in a peer-to-peer manner may be required, the major applications with M2M communication networks require the MTC devices to communicate with MTC servers in other network domains [3]. The Third Generation Partnership Project (3GPP) is active in developing M2M-related standards for Long Term Evolution (LTE) networks. According to [4], an MTC device is a user equipment (UE) for M2M communications. Using LTE networks as the air interface for M2M communications has several advantages [5]. The wide network coverage of LTE networks makes it possible to serve MTC devices in most urban and rural areas. The backhaul network of LTE networks can provide seamless communications between MTC devices and M2M application servers.

However, since LTE networks are optimized for human-to-human (H2H) communications, there are several problems when a large number of MTC devices try to access LTE networks. The first problem is efficiency. Compared with H2H communications which have high data rates, M2M communications usually feature low data rates as well as infrequent transmissions. The size of signalling packets in LTE networks to synchronize MTC devices to the evolved node B (eNodeB) or resolve contentions between MTC devices can be much larger than the size of user data packets for M2M applications [6]. The problem of low efficiency is even worse for battery-powered MTC devices since most of their limited power is used to transmit signaling packets. Another problem is congestion, including radio access network (RAN) congestion and core network (CN) congestion. The RAN congestion takes place when a large number of MTC devices attempt to access to an eNodeB. As described in [4], the number of MTC devices within a cell can be significantly large, *e.g.,* thousands of MTC devices accessing an eNodeB. The system suffers from severe congestion if these MTC devices try to access to the eNodeB within a short period of time. Congestion can also take place at the CN when packets from different eNodeBs try to access the same gateway node of

Manuscript received Apr. 21, 2014; revised Dec. 7, 2014, May 3, 2015, Oct. 2, 2015, and Dec. 4, 2015; accepted Jan. 20, 2016. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Part of this paper was presented at the *IEEE Global Communications Conference (GLOBECOM'13)*, Atlanta, GA, Dec. 2013. The review of this paper was coordinated by Prof. Yu Cheng.

Suyang Duan was with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. He is now with Viavi Solutions Canada (e-mail: suyangd@ece.ubc.ca).

Vahid Shah-Mansouri is with School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran (email: vmansouri@ut.ac.ir).

Zehua Wang and Vincent Wong are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: zwang@ece.ubc.ca; vincentw@ece.ubc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.XXXXXXX

the CN. According to the work in [7], congestion and traffic overload caused by MTC devices may be due to following reasons: a) massive number of MTC devices may be activated simultaneously by an external event; b) recurring MTC devices that are synchronized to communicate with M2M application servers together in a periodic manner. Depending on the network infrastructure, both the RAN congestion and the CN congestion may occur.

Many solutions are proposed by 3GPP to alleviate the traffic overload caused by MTC devices [4]. These solutions include separating or dynamically allocating the random access channels (RACHs) to MTC devices, or applying the slotted access scheme, the pull-based scheme, the MTC device backoff scheme, and the access class barring (ACB) scheme. Among these solutions, ACB is one of the most efficient methods. When an MTC device tries to initiate a transmission, it generates a random number between 0 and 1, and compares the generated number with the ACB factor broadcast by the eNodeB. If the number is less than the ACB factor, the MTC device proceeds to access the eNodeB. Otherwise, it needs to backoff temporarily.

In the literature, there are various works that study congestion control for M2M communications. When slotted ALOHA is used as the medium access control (MAC) scheme for MTC devices, there is an optimal number of MTC devices being allowed to access an eNodeB simultaneously so the throughput can be maximized. When the eNodeB notices that the number of MTC devices requiring access to it is more than the optimal number, the eNodeB uses an ACB factor to maintain the optimal number of MTC devices that access to it simultaneously. This scheme is discussed in [8], which uses the channel statistical occupancy rate to estimate the traffic load by monitoring the ratio of the number of busy channels over the number of all sampling channels. The scheme outperforms slotted ALOHA for high traffic load, which is a common scenario in M2M communications.

Lien *et al.* in [9] investigate the problem of how the ACB factors can be jointly determined among several neighbouring eNodeBs. They assume that the coverage of different eNodeBs has overlapped areas. The MTC devices located in the overlapped areas can choose one of the eNodeBs to access to. The scheme contains two steps. First, it provides a strategy for each MTC device to independently choose an eNodeB to connect with based on the default ACB factors broadcast by those eNodeBs. Then, given that eNodeBs have the information of the locations of all MTC devices as well as the strategies adopted by these MTC devices, the eNodeBs divide these MTC devices into each cell evenly and then update the ACB factors with the optimal values accordingly.

In [10], a congestion-aware admission control scheme is proposed to obtain the ACB factor. Instead of estimating the ACB factor from the traffic of the RAN, this factor is obtained based on the congestion level at the CN. The system uses a proportional integral derivative (PID) controller to adaptively change the ACB factor, using the difference between the current queue length and the reference value as the input. Compared to fixed ACB, the scheme can reduce the queue length and the number of dropped packets at the CN.

Using drift analysis, Wu *et al.* in [11] utilize the statistics of consecutive idle and collision slots to reduce access delay in a bursty traffic situation. As the number of contending nodes in random access has great influence on system performance, an algorithm is proposed to estimate the number of MTC devices that try to access to an eNodeB. The transmission probability of each device (*i.e.*, the ACB factor) can then be determined. Another algorithm based on drift analysis to estimate the number of backlogged MTC devices is proposed in [12]. In this algorithm, the backlog estimation is determined iteratively with a linear function by taking the number of successful transmissions, number of collisions, and number of idle channels into account. The eNodeB can then estimate the ACB factor and broadcast the result accordingly.

Sheu *et al.* in [13] propose an adaptive scheme to schedule MTC devices that need to periodically connect to an eNodeB. MTC devices inherit the same contention resources once they succeed in their first attempts so as to avoid collisions. These MTC devices will keep on using the same contention resources until the contention level at the eNodeB is stable. Then the eNodeB will reduce the resources allocated for MTC devices. Each MTC device will determine which resources to access accordingly based on a rule known to all MTC devices so that these rescheduled devices will not collide in accessing MTC resources.

In addition to ACB, the slotted access scheme has also been studied. Liu *et al.* in [14] propose a frame-based hybrid MAC scheme for M2M communication networks. In this scheme, a frame is divided into a contention period and a transmission period. The length of both periods can be changed dynamically. MTC devices first contend for transmission during the contention period. The transmission period provides access opportunities for the devices that succeed in the contention period. An optimization problem on how to set the durations of both periods is formulated to maximize the system throughput.

As MTC devices can support a wide range of applications which have different quality of service (QoS) requirements, congestion control schemes can be designed based on satisfying the QoS requirements of each class and allocating resources among different classes. In [15], a prioritized random access scheme is proposed to reduce RAN overload, which is achieved by pre-allocating RACH resources for different MTC classes with class-dependent backoff procedures. Within each class, the eNodeB continuously monitors the number of successful transmissions to decide whether the system is suffering from congestion. Simulation results show that the pre-allocating scheme achieves good performance to reduce the average access delay for MTC devices and satisfies the QoS requirements for MTC devices in different classes. Lien *et al.* in [16] study the QoS requirement in terms of packet delay. The network only allows access attempts from MTC devices within an allocated access grant time interval (AGTI). Different AGTIs are allocated to each class based on its access priority and traffic rate. Kwon *et al.* in [17] study the problem of minimizing resources allocated for MTC devices in a multicell system and consider QoS requirement in terms of the outage-probability of communication links. They consider not only the collisions caused by simultaneous packet

transmissions from the MTC devices within the same cell but also the interference from MTC devices in the neighbouring cells.

Other approaches have been proposed for RAN overload problem which are inherently different from what is specified in the 3GPP standard. In [18], Niyato *et al.* consider a heterogeneous cellular network with different types of eNodeBs (macrocells and small cells). Traffic in the macrocells is offloaded to the small cells in order to avoid congestion. They propose a queuing model to evaluate the performance of such network. In [19], Osti *et al.* apply a queuing system to model the arrival process of the contention resolution messages sent by the eNodeB. They further use a Markov model to analyze the performance of downlink control channels for random access in LTE networks.

In this paper, our focus lies in alleviating the RAN congestion in LTE networks. We aim to manage random access attempts at the side of MTC devices to reduce the congestion in an overloaded condition instead of rejecting access at the eNodeB or the CN. In case of an emergency, it is crucial that data from all MTC devices is collected as soon as possible. Therefore, we need to minimize the total amount of time it takes for all active MTC devices to finish transmitting user data packets for M2M applications. We consider the use of the ACB scheme with an *adaptive* ACB factor. The contributions of this paper are as follows:

- We first determine the minimum expected total service time, *i.e.*, the time required for all MTC devices to successfully access an eNodeB. The theoretical value is validated by simulations.
- We derive a lower bound on the amount of resource required to accommodate the random access for MTC devices in a fixed resource allocation (FRA) scheme. To reduce the amount of random access resources required by a large number of MTC devices, we also propose a dynamic resource allocation (DRA) scheme.
- We propose two dynamic access class barring (D-ACB) algorithms without the backlog information for FRA and DRA schemes, respectively. Both algorithms update the ACB factor by using the real time traffic information available to the eNodeB.
- Simulation results show that our proposed D-ACB algorithms can achieve close to the optimal performance as in the ideal case where the backlog information is available to the eNodeB. Results also show that D-ACB algorithm for FRA outperforms the scheme in [12] that uses the ACB factor determined by backlog estimation based on drift analysis. We further conduct simulations to compare the average random access resources required to serve each MTC device by using FRA and DRA schemes.

Our work differs from the related works in different directions. In our work, we use the number of collisions in the RAN to determine the ACB factor. This is different from [10] using the collision information in the CN, the algorithm in [8] which uses the channel statistical occupancy rate to estimate the traffic load and determine the ACB factor, and [15] which uses the number of successful transmissions to monitor the congestion level in the system. In addition, we formulate our problem with a multi-channel random access model. This is different from the single channel random access model used in [8] and [11]. Our work is also different in our beta distribution activation model instead of the conventional Poisson distribution, the latter of which is suitable for the activation pattern with an exponential inter-arrival time rather than a limited time bursty traffic.

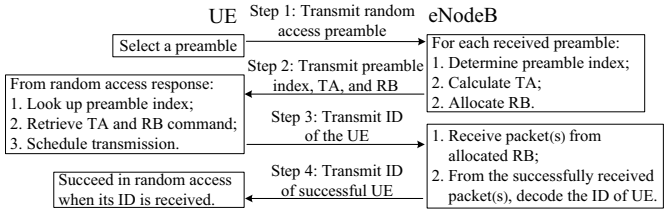


Fig. 1. The contention-based random access in LTE networks.

The rest of the paper is organized as follows. In Section II, we summarize the random access procedures in LTE networks. In Section III, we present the analytical model to determine the total service time. In Section IV, we determine the optimal ACB factor in the FRA scheme and then propose the DRA scheme. In Section V, we propose our D-ACB algorithms for FRA and DRA schemes to determine the ACB factors without *a priori* knowledge of the system backlog. In Section VI, performance evaluation is presented. The paper is concluded in Section VII.

II. Random Access Procedures in LTE Networks

In LTE networks, user data is scheduled to be transmitted through the Physical Uplink Shared CHannel (PUSCH). Asynchronous devices acquire synchronization with the eNodeB and reserve uplink channel using RACHs. RACHs are time-frequency resource blocks (RBs) repeated in the system periodically. There is a set of codes called *preambles* which are shared by all users in their random access. Each node requesting an uplink channel transmits a random access preamble in a RACH. There are two types of access modes in RACHs. The first one is contention-based, which is used for regular users. A user selects a preamble randomly from the set of available preambles. In this case, two nodes may select the same preamble. The second type is contention-free, where the preambles are assigned by the eNodeB. Thus, simultaneous transmissions of the same preamble by different nodes do not occur in the contention-free access mode. This provides low latency service for users with high priority (*e.g.*, handover). In this paper, we only focus on the contention-based access mode. Fig. 1 summarizes the contention-based random access mechanism in LTE networks, which consists of the following steps [20]:

In Step 1, each UE randomly selects a random access preamble from a pool of random access preambles known to both UEs and the eNodeB. The transmission of this preamble serves as a random access request to obtain a dedicated time-frequency RB for the upcoming scheduled data transmissions in Step 3. The UEs transmit their preambles but not their identifiers (IDs) in the random access request. Copies of

the same preamble may be received by the eNodeB when multiple UEs select and transmit the same preamble. In Step 2, the eNodeB acknowledges each of different preambles it has received with a random access response. Each random access response conveys the index of the preamble being acknowledged, the timing alignment (TA) instruction for UEs that have transmitted the preamble, and the RB allocation command for these UEs. In Step 3, a UE first finds its random access response by looking up the index of the preamble it has used in its random access request, and then uses the dedicated RB on PUSCH to transmit its ID to the eNodeB. If more than one UE has transmitted the same preamble in Step 1, they will be instructed to transmit their packets within the same time-frequency RB in Step 3. In this case, the packet collisions may occur at the eNodeB. For each packet that is successfully decoded in Step 3, which contains the ID of the receiver, a contention resolution message is sent to the corresponding UE in Step 4. Note that if packet collision occurs and the eNodeB is able to decode one of the collided packets in Step 3, it will acknowledge the UE whose message is successfully received. Unacknowledged UEs remain silent until the next RACH when they repeat the random access procedures starting from Step 1.

In LTE cellular networks, 64 preambles are available for random access, among which some are reserved for contention-free access. When MTC devices access an eNodeB in LTE cellular networks, they have to share the remaining preambles for contention-based access with H2H UEs, *e.g.*, smartphones. In our model, we assume that different resources are allocated to M2M traffic and H2H traffic. Hence, we consider how MTC devices compete for dedicated preambles among themselves only. Note that random access can only take place within certain time-frequency RBs specified by the eNodeB, *i.e.*, Physical Random Access CHannel (PRACH), which is the physical layer mapping of RACH. Depending on different configurations, RACHs can be scheduled very differently in terms of time and frequency. For example, when the PRACH configuration index is set to be 6, RACHs will occur every 5 ms within a bandwidth of 180 kHz with a duration ranging from 1 ms to 3 ms [21], [22]. In this paper, we only consider transmissions within random access channels. Here, the term *channel* refers to a time-frequency RB, not the medium that electromagnetic waves travel. In the following analysis, we will use the terms *channel* and ***RB*** interchangeably. Specifically, a PRACH is a periodic time-frequency RB, where random access attempts of MTC devices can take place.

III. System Model

Consider N MTC devices which have previously registered with an eNodeB. The eNodeB is aware of the total number of devices within its coverage. These devices have just recovered from an emergency, *e.g.*, a power blackout, and all of them try to re-establish synchronization with the eNodeB. As these devices are not synchronized, they will not be activated simultaneously but within a short period of time T_A. We refer to this period as the *activation time*. Each MTC device is activated at time t with probability density function $g(t)$. A

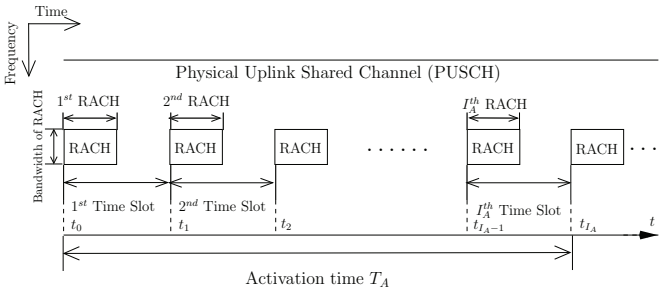


Fig. 2. Random access channels are time frequency resource blocks that repeat periodically. We divide the activation time into I_A time slots so that there is one random access channel during each time slot.

popular choice to model the burstiness of the traffic, proposed by 3GPP in [21], is the beta distribution

$$g(t) = \frac{t^{\alpha-1}(T_A - t)^{\beta-1}}{T_A^{\alpha+\beta-2}\mathcal{B}(\alpha,\beta)}, \quad 0 \le t \le T_A, \tag{1}$$

where $\mathcal{B}(\alpha,\beta)$ is the beta function [23].

We denote I_A as the number of random access channels within the activation time. We divide the activation time into I_A discrete slots, each of which is longer than the duration of a random access channel. Slot i ($i = 1, 2, \ldots$) begins with the i^{th} random access channel, as shown in Fig. 2. The length of each time slot is equal to the interval between two consecutive random access channels. The i^{th} time slot starts at time t_{i-1} and ends at time t_i. The first time slot starts from $t_0 = 0$. The last one ends at $t_{I_A} = T_A$. To simplify the model, we assume that new activations within time slot i, *i.e.*, within $[t_{i-1}, t_i]$, will only take place at the beginning of this time slot and choose the random access channel in this time slot for their first random access attempts. We denote by λ_i the number of new activations (arrivals) during time slot i, for $i = 1, 2, \ldots, I_A$. According to [21], λ_i is subject to the distribution of activation traffic $g(t)$ and the total number of devices N. Specifically, we have

$$\lambda_i = \left\lceil N \int_{t_{i-1}}^{t_i} g(t)\mathrm{d}t \right\rceil, \quad i = 1, 2, \ldots, I_A, \tag{2}$$

where the ceiling function $\lceil \cdot \rceil$ is used to ensure that λ_i is an integer.

In order to alleviate random access congestion, the eNodeB broadcasts an ACB factor p as part of the system information before each random access opportunity using System Information Block (SIB). In each time slot, an MTC device, which has not yet connected to the network, generates a random number between 0 and 1. If this number is less than p, then the MTC device selects a random access preamble and sends the preamble to the eNodeB. Otherwise, the MTC device stays silent and waits for the random access channel in the next time slot. These backlogged users as well as newly activated users will perform the ACB check before transmitting random access preambles to the eNodeB in the next time slot (random access channel). If multiple MTC devices select the same preamble, then the packet collision will occur at the eNodeB in Step 3 as we explained in Fig. 1. We assume that when a packet

collision occurs, the eNodeB is not able to decode any message from the collided packets. Thus, none of the MTC devices that suffer from the packet collision succeeds to access to eNodeB in this time slot. Whenever a user fails in one time slot, it will try to select and send a preamble sequence in the following time slots after passing the ACB check. This scheme uses the deferred first transmission, where new arrivals are treated as backlogged users.

We are interested in estimating the total time it takes for the eNodeB to collect all user data packets. After an MTC device transmits a preamble successfully, its user data packet can be transmitted without contention via the scheduled RBs on PUSCH, which takes a constant time. Therefore, the dominant part is the time for all MTC devices to successfully transmit their preamble sequences, which is referred to as the *total service time* in this paper. In our model, it takes the system I_X time slots before all preamble sequences of MTC devices are successfully transmitted. As I_X is a random variable, we propose to determine its expectation, *i.e.*, $\mathbb{E}[I_X]$, in the following discussions.

For the i^{th} random access channel (*i.e.*, the random access channel in the i^{th} time slot), we introduce an $(N+1)\times 1$ state vector $\boldsymbol{q}_i = (q_{i,0}, q_{i,1}, \ldots, q_{i,N})$, which represents the probability distribution of the backlog in the system at time slot i. Specifically, the element $q_{i,n}$ in state vector $\boldsymbol{q}_i$ denotes the probability that there are n backlogged users right after time slot i. By definition, $\sum_{n=0}^{N} q_{i,n} = 1$, for $i = 0, 1, \ldots$. At the first time slot starting at time $t_0 = 0$, we have $q_{0,0} = 1$ and $q_{0,n} = 0$, for $n = 1, 2, \ldots, N$.

When $i > I_A$, no more new activation takes place. The probability that there is no backlog at $i = (I_A + 1)$ can be zero. As i increases in the system, $q_{i,0}$ starts growing and eventually approaches 1. Let $\hat{i}$ denote the smallest $i > I_A$ such that the value of $q_{i,0}$ (*i.e.*, the probability that we have zero backlog in the system at random access slot i) is positive. That is,

$$\hat{i} = \min_{i=0,1,2,\ldots} \{i\} \text{ subject to } q_{i,0} > 0,\ i > I_A. \tag{3}$$

For $i > \hat{i}$, by definition, $q_{i-1,0}$ and $q_{i,0}$ denote the probability that there is no backlog in the system at the beginning and at the end of time slot i, respectively. The probability that the system finishes all transmissions at time slot i is $(q_{i,0} - q_{i-1,0})$. The expectation of I_X is given by

$$\mathbb{E}[I_X] = \sum_{i=1}^{\infty} i(q_{i,0} - q_{i-1,0}). \tag{4}$$

As $q_{i,0} = 0$, for $i = 1, 2, \ldots, \hat{i} - 1$, equation (4) becomes

$$\mathbb{E}[I_X] = \hat{i} q_{\hat{i},0} + \sum_{i=\hat{i}+1}^{\infty} i(q_{i,0} - q_{i-1,0}). \tag{5}$$

We now determine how the value of $q_{i,0}$ evolves with time (*i.e.*, as i increases). We divide the state transition within each time slot into two steps: the new activation step and the transmission step. At the end of time slot $i-1$, the distribution of the backlog is given by the state vector $\boldsymbol{q}_{i-1}$. Specifically, we have

$$\boldsymbol{q}_{i-1} = (q_{i-1,0}, q_{i-1,1}, q_{i-1,2}, \ldots, q_{i-1,z}, 0, 0, \ldots, 0), \tag{6}$$

where z is the largest possible backlog at the end of time slot $i-1$. This means the probability that there are more than z backlogged UEs in the system is zero. During time slot i, there are λ_i newly activated MTC devices ($\lambda_i = 0$ for $i > I_A$). We take the newly activated users into account by right shifting the first z elements in vector $\boldsymbol{q}_{i-1}$ with λ_i units while keeping its dimension $((N+1)\times 1)$ unchanged. After the shifting, We have

$$\begin{aligned}\boldsymbol{q}'_i \triangleq (\underbrace{0, 0, \ldots, 0}_{\lambda_i}, q_{i-1,\lambda_i},\\ q_{i-1,\lambda_i+1}, q_{i-1,\lambda_i+2}, \ldots, q_{i-1,\lambda_i+z}, 0, 0, \ldots, 0).\end{aligned} \tag{7}$$

Specifically, when $\lambda_i = 0$, no shifting is necessary. The state vector $\boldsymbol{q}'_i$ is the probability distribution of the backlog after the activation step of time slot i but before the step of transmitting random access preambles.

To model the transmission step, we introduce the following $(N+1)\times(N+1)$ transition matrix

$$\boldsymbol{R} = \begin{pmatrix} r_{0,0} & r_{1,0} & \cdots & r_{N,0} \\ r_{0,1} & r_{1,1} & \cdots & r_{N,1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{0,N} & r_{1,N} & \cdots & r_{N,N} \end{pmatrix}, \tag{8}$$

where the element $r_{s,t}$ is the probability that the backlog changes from s to t. As a result, the probability distribution of the backlog at the end of time slot i is given by $\boldsymbol{q}_i = \boldsymbol{R}\boldsymbol{q}'_i$. Note that $r_{s,t} = 0$ for $t > s$ since we consider that no MTC device is activated during the transmission step.

Now we derive the expression for $r_{s,t}$. When the backlog changes from s to t, it means that among the s backlogged users, $s - t$ users transmit their random access preambles successfully. Let random variables K_i and N_i denote the number of successfully preamble transmissions and the number of users requiring access to eNodeB in time slot i, respectively. Note that there are M preambles available in the system. We now determine the probability $\mathbb{P}(K_i = k \mid N_i = n)$, which is the probability that there are $K_i = k$ ($k \le M$) preambles successfully transmitted given the event that $N_i = n$ users require access to the eNodeB in time slot i. This probability consists of the following three parts:

1) Among n backlogged users, there are $N_i^a = j$ users who pass the ACB check and transmit their preambles, $\mathbb{P}(N_i^a = j \mid N_i = n)$. The relation between $\mathbb{P}(N_i^a = j \mid N_i = n)$ and $\mathbb{P}(K_i = k \mid N_i = n)$ is as follows:

$$\begin{aligned}\mathbb{P}(K_i = k \mid N_i = n) = \sum_{j=0}^{n} \mathbb{P}(N_i^a = j \mid N_i = n)\\ \times \mathbb{P}(k \text{ successful tranmissions} \mid j \text{ transmissions}).\end{aligned}$$

2) Among j transmitted preambles, k preambles succeed.
3) The remaining $j - k$ preambles collide.

The first part can be obtained as

$$\mathbb{P}(N_i^a = j \mid N_i = n) = \binom{n}{j} p^j (1-p)^{n-j}, \quad j = 0, 1, \ldots, n. \tag{9}$$

An analogy of the second and third parts would be to place j different objects into M different cells, on condition that there are k cells that have one object in each of them, and the remaining cells have either no object or at least two objects. The number of ways of putting j different objects into M different cells is M^j. First, we choose k objects and k cells, and put one object in each cell. The number of different combinations is $\binom{j}{k}\binom{M}{k}k!$. Then, we put the remaining $j-k$ objects into the remaining $M-k$ cells so that each of these $M-k$ cells either has no object or at least two objects in it. We refer to the number of different ways as $f(j-k, M-k)$. When M is equal to k, then there is no cell to put any objects, so that $f(j-k, 0) = 0$, $j \neq k$. When j is equal to k, we have $f(0,0) = 1$.

We denote $S_l,\ l = 1, 2, \ldots, M-k$ as the set of events, where the l^{th} cell has exactly one object. Then, the set $S = S_1 \cup S_2 \cup \cdots \cup S_{M-k}$ includes all the cases that at least one cell has exactly one object. Using the principle of inclusion and exclusion [24], the cardinality of this set is

$$\begin{aligned}
|S| &= |S_1 \cup S_2 \cup \cdots \cup S_{M-k}| \\
&= (-1)^0 \sum_{l=1}^{M-k} |S_l| + (-1)^1 \sum_{l=1}^{M-k} \sum_{r \neq l} |S_l \cap S_r| \\
&\quad + (-1)^2 \sum_{l=1}^{M-k} \sum_{r \neq l} \sum_{\substack{v \neq l \\ v \neq r}} |S_l \cap S_r \cap S_v| \\
&\quad + \cdots + (-1)^{M-k-1} |S_1 \cap S_2 \cap \cdots \cap S_{M-k}|,
\end{aligned} \tag{10}$$

in which

$$\begin{aligned}
&(-1)^0 \sum_{l=1}^{M-k} |S_l| \\
&= (-1)^0 \binom{M-k}{1} \binom{j-k}{1} 1! (M-k-1)^{j-k-1},
\end{aligned} \tag{11}$$

and

$$\begin{aligned}
&(-1)^1 \sum_{l=1}^{M-k} \sum_{r \neq l} |S_l \cap S_r| \\
&= (-1)^1 \binom{M-k}{2} \binom{j-k}{2} 2! (M-k-2)^{j-k-2}.
\end{aligned} \tag{12}$$

If $(M-k) < (j-k)$, then the last term in (10) is $(-1)^{M-k-1}|S_1 \cap S_2 \cap \cdots \cap S_{M-k}|$. Otherwise, the last $M-j$ terms do not exist, and the term will be $(-1)^{j-k-1}|S_1 \cap S_2 \cap \cdots \cap S_{j-k}|$. We denote $u \triangleq \min(M-k, j-k)$. Then, the last term of this series will be

$$\begin{aligned}
&(-1)^{u-1} |S_1 \cap S_2 \cap \cdots \cap S_u| \\
&= (-1)^{u-1} \binom{M-k}{u} \binom{j-k}{u} u! (M-k-u)^{j-k-u}.
\end{aligned} \tag{13}$$

Therefore,

$$|S| = \sum_{l=1}^{u} (-1)^{l-1} \binom{M-k}{l} \binom{j-k}{l} l! (M-k-l)^{j-k-l}. \tag{14}$$

Our goal is to determine the total number of cases where no cell has exactly one object in it, which is the cardinality of set $\overline{S}$.

$$\begin{aligned}
|\overline{S}| &= (M-k)^{j-k} - |S| \\
&= (M-k)^{j-k} \\
&\quad + \sum_{l=1}^{u} (-1)^l \binom{M-k}{l} \binom{j-k}{l} l! (M-k-l)^{j-k-l} \\
&= \sum_{l=0}^{u} (-1)^l \binom{M-k}{l} \binom{j-k}{l} l! (M-k-l)^{j-k-l} \\
&= f(j-k, M-k).
\end{aligned} \tag{15}$$

Thus,

$$\begin{aligned}
&\mathbb{P}(K_i = k \mid N_i = n) \\
&= \sum_{j=0}^{n} \mathbb{P}(N_i^a = j \mid N_i = n) \frac{\binom{j}{k}\binom{M}{k} k! f(j-k, M-k)}{M^j} \\
&= \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} \binom{j}{k} \binom{M}{k} k! \\
&\quad \times \frac{\sum_{l=0}^{u} (-1)^l \binom{M-k}{l} \binom{j-k}{l} l! (M-k-l)^{j-k-l}}{M^j}, \\
&\quad k = 0, 1, \ldots, M.
\end{aligned} \tag{16}$$

Note that the value of the element $r_{s,t}$, for $s = 0, \ldots, N$ and $t = 0, \ldots, N$, in matrix $\boldsymbol{R}$ defined by (8) is equal to $\mathbb{P}(K_i = s - t \mid N_i = s)$. Thus, with (16), we can determine the transition matrix $\boldsymbol{R}$ and $\boldsymbol{q}_i$ for each time slot i. The value of $\mathbb{E}[I_X]$ can thus be determined by (5).

IV. Optimal Scenarios with Full State Information

In this section, we assume that the eNodeB has full state information, *i.e.*, the actual number of backlogged users (*i.e.*, the number of MTC devices that are requiring access to the eNodeB) in each time slot is available to the eNodeB. We first derive the optimal ACB factor, *i.e.*, the value of p. We then investigate a dynamic resource allocation scheme based on the full state information.

A. Optimal ACB Factor

The ACB factor p plays an important role in the performance of contention control in a random access channel. Therefore, it is of interest to find the optimal value of p. By definition, p is the probability that a user passes ACB check, *i.e.*, $p = \mathbb{P}(N_i^a = 1 \mid N_i = 1)$. We consider $N_i^a = j$ users among $N_i = n$ backlog pass the ACB check. We further consider each of these users that passed the ACB check selects a random access preamble from M available preambles with an equal probability given by $\frac{1}{M}$. For a given preamble m

transmitted to the eNodeB, let $D_m = 0, 1, c$ denote the cases that the preamble m is selected by none of the users, by exactly one user, and by more than one user, respectively. The probability that only one user selects preamble m is

$$\mathbb{P}(D_m = 1 \mid N_i^a = j) = \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1}. \quad (17)$$

The expected number of successful preamble transmissions in time slot i is given by

$$\begin{aligned} \mathbb{E}\left[K_i \mid N_i^a = j\right] &= \sum_{m=1}^{M} \mathbb{P}(D_m = 1 \mid N_i^a = j) \\ &\quad \times M \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1}. \end{aligned} \quad (18)$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[K_i \mid N_i = n\right] &= \sum_{j=1}^{n} \mathbb{P}(N_i^a = j \mid N_i = n) \\ &\quad \times M \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= \sum_{j=1}^{n} \binom{n}{j} p^j (1-p)^{n-j} j \Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= np\Big(1 - \frac{p}{M}\Big)^{n-1}. \end{aligned} \quad (19)$$

The minimum expected total service time can be achieved when the expected number of successful preamble transmissions in each time slot is maximized. By taking the derivative of (19) with respect to p, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}p}\mathbb{E}(K_i \mid N_i = n) = n\Big(1 - \frac{p}{M}\Big)^{n-2}\Big(1 - \frac{np}{M}\Big). \quad (20)$$

When $M \geq n$, we have $\frac{\mathrm{d}}{\mathrm{d}p}\mathbb{E}(K_i \mid N_i = n) \geq 0$. The maximum throughput is achieved when $p = 1$. In other words, when the preamble number is larger than the number of backlogged users, the ACB factor should be set to 1. In this situation, no ACB check will be performed and packets will be transmitted upon activation. When $M < n$, we set $\frac{\mathrm{d}}{\mathrm{d}p}\mathbb{E}(K_i \mid N_i = n) = 0$, and obtain $p = \frac{M}{n}$. Therefore, we have

$$p^\star = \min\left(1, \frac{M}{n}\right). \quad (21)$$

By dynamically updating the ACB factor during each time slot according to (21), the minimum total service time can be achieved. We refer to this scheme as the *optimal p scenario*. The figures to be presented in Section VI show that the analytical and simulation results under the optimal p scenario match with each other exactly.

B. Dynamic Allocation of Preambles for M2M Traffic

In the previous sections, we focused on fixed resource allocation (FRA) scheme, *i.e.*, during each time slot, the number of random access preambles allocated to MTC devices is fixed. In LTE networks, each cell has a limited number of 64 preambles. These random access resources are shared by H2H UEs and MTC devices. We refer to each random access preamble used in each time slot as a *random access opportunity*. For the FRA scheme, the system allocates M opportunities in one time slot for I_X time slots, and the average number of opportunities per MTC device is $\frac{1}{N} M I_X$. We notice that I_X is a function of N, M, and the ACB factor p. In general, I_X is an increasing function of N and a decreasing function of M. We investigate the relation between the average number of opportunities required to serve each MTC device and the values of N and M by simulations in Section VI.

Instead of dedicating a fixed number of preambles to MTC devices, the system can potentially change the number of preambles allocated to them over time based on traffic load. We denote the number of preambles allocated for MTC devices in time slot i as M_i. If the resources are dynamically allocated, then the average number of opportunities per MTC device is $\frac{1}{N}\sum_{i=1}^{I_X} M_i$. In this subsection, we discuss whether this value can be reduced by allocating preambles to MTC devices dynamically. Since MTC devices share random access resources with UEs in H2H communications, reducing the resources consumed by MTC devices will result in more resources available to the UEs in H2H communications such as the smartphones. From the perspective of the service provider, it is desirable to use fewer resources to serve MTC devices so as to accommodate more smartphone users, which are more profitable.

During each time slot, the conditional probability of a preamble being selected by exactly one user is

$$\mathbb{P}(D_m = 1 \mid N_i = n) = \binom{n}{1} \frac{p}{M} \Big(1 - \frac{p}{M}\Big)^{n-1}.$$

When p takes the optimal value, *i.e.*, $p = \frac{M}{n}$, we have

$$\mathbb{P}(D_m = 1 \mid N_i = n) = \binom{n}{1} \frac{1}{n} \Big(1 - \frac{1}{n}\Big)^{n-1} = \Big(1 - \frac{1}{n}\Big)^{n-1}. \quad (22)$$

For large values of n, we have $\lim_{n\to\infty} \mathbb{P}(D_m = 1 \mid N_i = n) = e^{-1}$. This limit holds in practice, as the number of MTC devices within a cell tends to be very large. This shows that one random access opportunity can accommodate e^{-1} successful random access attempt on average.

As we assume that the eNodeB knows the current backlog in the system, we can design a simple rule for the eNodeB to update the number of preambles allocated to MTC devices. By using M preambles, we can successfully accommodate Me^{-1} users in a time slot on average. Therefore, there is a linear relation between the number of preambles and the number of successful preamble transmissions. In light of this, we dynamically update the number of preambles M_i in time slot i to be proportional to the backlog n, which is given by

$$M_i = \min\left\{\left\lceil \frac{n}{b} \right\rceil, 64\right\}, \quad (23)$$

where b is a design parameter and remains a constant throughout the operation time of the system. The range of this parameter is estimated based on the number of MTC devices as well as the activation time to make sure that M_i will not exceed beyond its upper bound as the number of preambles within each cell is limited to 64.

Note that in practice, the eNodeB may not be able to obtain the backlog information of the system. Thus the optimal p scenario can only serve as a reference for the theoretical minimum total service time. In the following section, we propose a heuristic algorithm, which can adaptively update p to reduce the total service time. This algorithm is based on the information available to the eNodeB, so the D-ACB algorithm can be realized in practical systems. Simulation results on the algorithm with both FRA and DRA scheme will be presented in Section VI.

V. D-ACB: Dynamic Access Class Barring

In this section, we present the D-ACB algorithm to adaptively update the ACB factor p. As shown in (21), the optimal ACB factor depends on the backlog in the system, which is updated for each RACH. However, in practice, the backlog information in each time slot may not be available to the eNodeB. The available information is limited to the number of available preambles, number of successful preamble transmissions, number of unused preambles, and the total number of MTC devices registered in the system. In this section, we propose algorithms to dynamically adjust the ACB factors based on the available information listed above for FRA and DRA, respectively. Our work is based on the assumption that the eNodeB is not able to count the number of UEs that have transmitted the same preamble in a RACH when the preamble experiences collision. This is a widely adopted assumption which can also be found in [11], [12], and [25].

A. D-ACB with FRA

For a fixed number of preambles M, we first derive the expected number of preambles experiencing collisions during one RACH. At time slot i, consider that there are $N_i = n$ MTC devices which select one of the M preambles with equal probability. In the following analysis, we assume that $n \geq M$. When $n < M$, no ACB check will be performed and all the MTC devices will attempt random access upon activation. The probability for a backlogged user to pass the ACB check is p. The probability that preamble m is selected by a user in a time slot is thus $\frac{p}{M}$. The conditional probability that no user chooses preamble m is

$$\mathbb{P}(D_m = 0 \mid N_i = n) = \left(1 - \frac{p}{M}\right)^n. \tag{24}$$

The conditional probability that preamble m is selected by exactly one user is

$$\mathbb{P}(D_m = 1 \mid N_i = n) = \binom{n}{1} \frac{p}{M} \left(1 - \frac{p}{M}\right)^{n-1}. \tag{25}$$

Therefore, the conditional probability that preamble m is selected by more than one user (*i.e.*, the conditional probability that preamble m encounters collision when n MTC devices attempt random access in a time slot) is given by

$$\begin{aligned}\mathbb{P}(D_m = c \mid N_i = n) = 1 &- \mathbb{P}(D_m = 0 \mid N_i = n) \\ &- \mathbb{P}(D_m = 1 \mid N_i = n).\end{aligned} \tag{26}$$

Let random variable $C_{M,p}$ denote the number of preambles experiencing collisions with ACB factor p when M preambles are available. The expected value of $C_{M,p}$ is given by

$$\begin{aligned}\mathbb{E}[C_{M,p}] &= \sum_{m=1}^{M} \mathbb{P}(D_m = c \mid N_i = n) \\ &= M\left(1 - \left(1 - \frac{p}{M}\right)^n - \frac{np}{M}\left(1 - \frac{p}{M}\right)^{n-1}\right).\end{aligned} \tag{27}$$

If p is equal to the optimal value $p^\star = \frac{M}{n}$, we obtain

$$\mathbb{E}[C_{M,p^\star}] = M\left(1 - \left(1 - \frac{1}{n}\right)^n - \left(1 - \frac{1}{n}\right)^{n-1}\right). \tag{28}$$

For large values of n, we have $\mathbb{E}[C_{M,p^\star}] \approx M(1-2e^{-1})$ since $\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n = \lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{(n-1)} = e^{-1}$. We denote $\mu_{M,p^\star}$ as the average number of preambles experiencing collisions in the RACH with optimal ACB factor $p^\star$ when M preambles are available. Thus, when the value of n is large, we have the approximation

$$\mu_{M,p^\star} \approx M(1 - 2e^{-1}). \tag{29}$$

As mentioned earlier, for n backlogged MTC devices requiring access to the eNodeB in a RACH, the optimal ACB factor is $p^\star = \frac{M}{n}$. However, n may not be available to the eNodeB. We denote by $\widehat{n}$ an estimate of the actual number of backlogged MTC devices n, and use this estimate to obtain a sub-optimal ACB factor $\widehat{p} = \frac{M}{\widehat{n}}$. Let random variable $C_{M,\widehat{p}}$ denote the number of preambles experiencing collisions in a RACH with the ACB factor $\widehat{p} = \frac{M}{\widehat{n}}$ where there are n backlogged MTC devices and M preambles are available. By substituting $p = \widehat{p} = \frac{M}{\widehat{n}}$ into (27), the expected value of random variable $C_{M,\widehat{p}}$ is given by

$$\begin{aligned}\mathbb{E}[C_{M,\widehat{p}}] &= M\left(1 - \left(1 - \frac{1}{\widehat{n}}\right)^n - \frac{n}{\widehat{n}}\left(1 - \frac{1}{\widehat{n}}\right)^{n-1}\right) \\ &\approx M\left(1 - e^{-\frac{n}{\widehat{n}}} - \frac{n}{\widehat{n}} e^{-\frac{n-1}{\widehat{n}}}\right).\end{aligned} \tag{30}$$

For large values of n, we have

$$\begin{aligned}\mathbb{E}[C_{M,\widehat{p}}] - \mu_{M,p^\star} &\approx M\left(1 - e^{-\frac{n}{\widehat{n}}} - \frac{n}{\widehat{n}} e^{-\frac{n}{\widehat{n}}}\right) - M(1 - 2e^{-1}) \\ &= 2Me^{-1}\left(1 - e^{-\frac{n-\widehat{n}}{\widehat{n}}} - \frac{n-\widehat{n}}{2\widehat{n}} e^{-\frac{n-\widehat{n}}{\widehat{n}}}\right).\end{aligned} \tag{31}$$

Assuming that the estimate of the number of backlogged MTC devices is not far from its actual value relatively (*i.e.*, $\frac{n-\widehat{n}}{\widehat{n}} \ll 1$), we can ignore the last term in (31). Approximating $e^{-\frac{n-\widehat{n}}{\widehat{n}}}$ by $1 - \frac{n-\widehat{n}}{\widehat{n}}$, we obtain

$$\mathbb{E}[C_{M,\widehat{p}}] - \mu_{M,p^\star} \approx \frac{2M\,(n - \widehat{n})}{\widehat{n}e}. \tag{32}$$

Therefore, the number of backlogged MTC devices n can be approximated by

$$n \approx \widehat{n}\left(1 + \frac{(\mathbb{E}[C_{M,\widehat{p}}] - \mu_{M,p^\star})\,e}{2M}\right). \tag{33}$$

Using this new estimate for the number of backlogged MTC devices, we can approximate the optimal ACB factor as

$$p^\star \simeq \min\left\{1, \widehat{p}\Big(1+\frac{(\mathbb{E}[C_{M,\widehat{p}}]-\mu_{M,p^\star})\,e}{2M}\Big)^{-1}\right\}. \quad (34)$$

In (34), we used the expected value of the random variable $C_{M,\widehat{p}}$. Due to the stochastic behavior of $C_{M,\widehat{p}}$, which is the result of changes in the number of backlogged devices, the expectation $\mathbb{E}[C_{M,\widehat{p}}]$ cannot be obtained. Instead, we use the observed number of collisions in each RACH as a measure of $\mathbb{E}[C_{M,\widehat{p}}]$. Let $\widehat{p}^{(i)}$ denote the ACB factor used in the i^{th} RACH and let $c^{(i)}_{M,\widehat{p}}$ denote the instance of random variable $C_{M,\widehat{p}}$ (*i.e.*, the observed number of collisions in the i^{th} RACH). Using the observed number of collisions $c^{(i)}_{M,\widehat{p}}$, we can improve our estimate of the optimal ACB factor in RACH i. Let $\widetilde{p}^{(i)}$ denote the improved ACB factor for i^{th} RACH based on the observation $c^{(i)}_{M,\widehat{p}}$. By substituting $c^{(i)}_{M,\widehat{p}}$ for $\mathbb{E}[C_{M,\widehat{p}}]$ and $\widetilde{p}^{(i)}$ for $p^\star$ in (34), we have

$$\widetilde{p}^{(i)} = \min\left\{1, \widehat{p}^{(i)}\Big(1+\frac{(c^{(i)}_{M,\widehat{p}}-\mu_{M,p^\star})e}{2M}\Big)^{-1}\right\}. \quad (35)$$

That is, we obtain the improved ACB factor $\widetilde{p}^{(i)}$ for the i^{th} RACH based on the observed number of collisions that happen in the same slot. The updated ACB factor $\widetilde{p}^{(i)}$ can be used to improve the estimate of the number of backlogged devices that attempted the random access in the i^{th} RACH. Specifically, let $\widetilde{n}^{(i)} \triangleq \frac{M}{\widetilde{p}^{(i)}}$ denote the improved estimate for the number of backlogged devices that attempted the random access in the i^{th} RACH. We use $\delta^{(i+1)}$ to represent the net change between the improved estimates of the number of backlogged devices in the i^{th} and $(i+1)^{\text{th}}$ RACHs. That is, $\delta^{(i+1)} \triangleq \widetilde{n}^{(i+1)} - \widetilde{n}^{(i)}$ for $i = 0, 1, 2\cdots$, where $\widetilde{n}^{(0)} = 0$. The value of $\delta^{(i+1)}$ depends on both the number of successfully served devices in the i^{th} RACH and the number of newly activated devices for the $(i+1)^{\text{th}}$ RACH. In reality, the time between two consecutive RACHs for random access in LTE varies from 1 ms to 20 ms [26], which means the number of newly activated devices between two consecutive time slots is small. Meanwhile, the number of successfully served devices is limited by the number of preambles and does not change significantly in two consecutive time slots as the ACB factor is gradually updated. Moreover, we have $\delta^{(i+1)} \geq -M$ for $i \geq 0$. The equality holds when each preamble is selected by exactly one MTC device in the i^{th} RACH (*i.e.*, M MTC devices are successfully served) and no MTC device is newly activated for the $(i+1)^{\text{th}}$ RACH. Based on these facts, the net change of the number of backlogged MTC devices in consecutive RACHs can be considered small. That is, we have $\delta^{(i+1)} \approx \delta^{(i)}$ for $i = 1, 2, \cdots$. By considering the improved estimate for the number of backlogged devices in the i^{th} RACH $\widetilde{n}^{(i)}$, we estimate the number of backlogged MTC devices for the $(i+1)^{\text{th}}$ RACH as follows:

$$\begin{aligned}\widehat{n}^{(i+1)} &= \max\left\{0, \widetilde{n}^{(i)} + \delta^{(i+1)}\right\} \\ &\approx \max\left\{0, \widetilde{n}^{(i)} + \delta^{(i)}\right\} \\ &= \max\left\{0, \widetilde{n}^{(i)} + \max\{-M, \widetilde{n}^{(i)} - \widetilde{n}^{(i-1)}\}\right\}.\end{aligned} \quad (36)$$

Algorithm 1 Algorithm for D-ACB with FRA

1: Input: N, M.
2: Initialization $\mu_{M,p^\star} := M(1-2e^{-1})$, $i := 0$, $W^{(0)} := 0$, $\widetilde{n}^{(0)} := 0$, $\widehat{p}^{(1)} := 1$.
3: **while** cumulative successful transmission $W^{(i)} < N$ **do**
4: Time slot $i := i + 1$.
5: Use ACB factor $\widehat{p}^{(i)}$ in the i^{th} RACH.
6: Monitor the number of successful transmissions: K_i.
7: Monitor the number of preambles that are chosen by more than one user: $c^{(i)}_{M,\widehat{p}}$.
8: Update $W^{(i)} := W^{(i-1)} + K_i$.
9: $\widetilde{p}^{(i)} := \min\left\{1, \widehat{p}^{(i)}\Big(1+\frac{(c^{(i)}_{M,\widehat{p}}-\mu_{M,p^\star})e}{2M}\Big)^{-1}\right\}$, $\widetilde{n}^{(i)} := \frac{M}{\widetilde{p}^{(i)}}$.
10: $\widehat{n}^{(i+1)} := \left\{0, \widetilde{n}^{(i)} + \max\{-M, \widetilde{n}^{(i)} - \widetilde{n}^{(i-1)}\}\right\}$.
11: $\widehat{p}^{(i+1)} := \min\left\{1, \frac{M}{\widehat{n}^{(i+1)}}\right\}$.
12: **end while**

Thus, the ACB factor $\widehat{p}^{(i+1)}$ used in the $(i+1)^{\text{th}}$ RACH is calculated by $\widehat{p}^{(i+1)} = \min\{1, \frac{M}{\widehat{n}^{(i+1)}}\}$.

Our proposed D-ACB with FRA algorithm is given in Algorithm 1. After taking the number of registered MTC devices N and the number of available preambles M as input (Step 1), the value of $\mu_{M,p^\star}$ is calculated (Step 2). The cumulative number of MTC devices which have successfully gained access up to RACH i is denoted by $W^{(i)}$ and is initialized by $W^{(0)} := 0$. The improved estimation for the number of backlogged MTC devices is initialized by $\widetilde{n}^{(0)} := 0$ and the ACB factor used for the 1^{st} RACH $\widehat{p}^{(1)}$ is initialized to 1 (Step 2). The loop (Steps 3-12) runs until all the MTC devices are successfully served. During RACH i (Step 4), the ACB factor $\widehat{p}^{(i)}$ is used (Step 5). The number of successful preamble transmissions K_i and the number of preambles selected by more than one user $c^{(i)}_{M,\widehat{p}}$ (*i.e.*, the number of preambles having collisions) are set by monitoring the RACH operation (Steps 6 and 7). K_i is used to update $W^{(i)}$ (Step 8). The value of $c^{(i)}_{M,\widehat{p}}$ is used to calculate the improved ACB factor $\widetilde{p}^{(i)}$, which is used to calculate the improved estimation for the number of backlogged MTC device $\widetilde{n}^{(i)}$ (Step 9). The backlog estimation for the $(i+1)^{\text{th}}$ time slot is determined according to (36), which is used to calculate the ACB factor for the $(i+1)^{\text{th}}$ RACH (Step 11).

We would like to highlight that Algorithm 1 determines the ACB factor for a time slot by considering the ACB factor that has been used for the previous RACH as well as the difference between the observed number of collided preambles and its expectation given by (26)–(28). By conducting extensive simulations in Section VI-A, we will show that Algorithm 1 can achieve almost the same performance as the case when the number of backlogged MTC devices in each time slot is available.

B. D-ACB with DRA

In D-ACB with DRA, the number of preambles M_i changes in each time slot i. We denote random variable $C_{M_i,p}$ as the number of preambles having collisions when M_i preambles are allocated and ACB factor p is used. Moreover, let

$\mu_{M_i,p^\star}$ denote the expected number of preambles experiencing collision in the RACH where the optimal ACB factor $p^\star = \frac{M_i}{n}$ is used and M_i preambles are available. We have $\mu_{M_i,p^\star} \approx M_i(1-2e^{-1})$, which is similar to (29). By following the similar approach that has been used in the previous section for D-ACB with FRA, similar equations to (24)–(33) can be obtained where M_i is replaced by M. For the i^{th} RACH, we have

$$p^\star \simeq \min\left\{1, \widehat{p}\Big(1+\frac{(\mathbb{E}[C_{M_i,\widehat{p}}]-\mu_{M_i,p^\star})\,e}{2M_i}\Big)^{-1}\right\}, \quad (37)$$

where $\widehat{p} = \frac{M_i}{\widehat{n}}$ is the sub-optimal ACB factor determined based on the estimation $\widehat{n}$ for n, and $M_i = \min\left\{\lceil\frac{\widehat{n}}{b}\rceil, 64\right\}$ by substituting $\widehat{n}$ for n in (23). Using $c^{(i)}_{M_i,\widehat{p}}$ as a measure of $C_{M_i,\widehat{p}}$ in (37), the sub-optimal ACB factor $\widehat{p}^{(i)}$ used for the i^{th} RACH with M_i preambles is given as follows:

$$\widetilde{p}^{(i)} = \min\left\{1, \widehat{p}^{(i)}\Big(1+\frac{(c^{(i)}_{M_i,\widehat{p}}-\mu_{M_i,p^\star})e}{2M_i}\Big)^{-1}\right\}. \quad (38)$$

Note that we have the improved estimate of the number of backlogged devices as $\widetilde{n}^{(i)} = \frac{M_i}{\widetilde{p}^{(i)}}$ for D-ACB with DRA. We use $\delta^{(i+1)} = \widetilde{n}^{(i+1)} - \widetilde{n}^{(i)}$ as the change of the improved estimate of the number of backlogged devices and assume that the changes in two consecutive time slots are almost the same. By substituting M_i for M into (36), we have

$$\widehat{n}^{(i+1)} \approx \big\{0, \widetilde{n}^{(i)} + \max\{-M_i, \widetilde{n}^{(i)} - \widetilde{n}^{(i-1)}\}\big\}. \quad (39)$$

We can now determine the optimal number of preambles which should be allocated to the backlogged MTC devices in the $(i+1)^{\text{th}}$ RACH by

$$M_{i+1} = \max\left\{1, \min\left\{\lceil\frac{\widehat{n}^{(i+1)}}{b}\rceil, 64\right\}\right\}. \quad (40)$$

The term 1 in (40) is used to avoid not allocating any preamble for M2M traffic in case there are very few devices in an RACH. The optimal ACB factor for the $(i+1)^{\text{th}}$ RACH is $\widehat{p}^{(i+1)} = \frac{M_{i+1}}{\widehat{n}^{(i+1)}}$. The term $\min\left\{\lceil\frac{\widehat{n}^{(i+1)}}{b}\rceil, 64\right\}$ is obtained by substituting $\widehat{n}^{(i+1)}$ for n in (23). Note that b is a system parameter. In Section VI-B, we further study the effect of parameter b on the performance of DRA by conducting simulations.

Our proposed D-ACB algorithm for DRA is given in Algorithm 2. The algorithm takes N and b as the input parameters (Step 1). The time slot index i, cumulative successful transmission $W^{(0)}$, and the improved backlog estimation $\widetilde{n}^{(0)}$ are initialized to 0, respectively (Step 2). Since the goal of DRA is to allocate fewer resources than FRA to service MTC devices, the number of preambles for the 1^{st} RACH is initialized by 1 and the value of $\mu_{M_1,p^\star}$ is calculated correspondingly (Step 2). The loop (Steps 3-13) keeps running until all MTC devices are successfully served. Steps 4-9 are similar to those in Algorithm 1. After estimating the number of backlogged MTC devices in the $(i+1)^{\text{th}}$ time slot (Step 10), the number of preambles M_{i+1} allocated for the $(i+1)^{\text{th}}$ RACH is calculated in Step 11. Then, the value of $\mu_{M_{i+1},p^\star}$ and the ACB factor used in the $(i+1)^{\text{th}}$ RACH are determined in Step 12, respectively. It should be noted that Algorithm 2 is designed for DRA where M_i preambles are used for the i^{th} RACH. It is different from Algorithm 1 where the number of preambles does not change in each time slot.

Algorithm 2 Algorithm for D-ACB with DRA

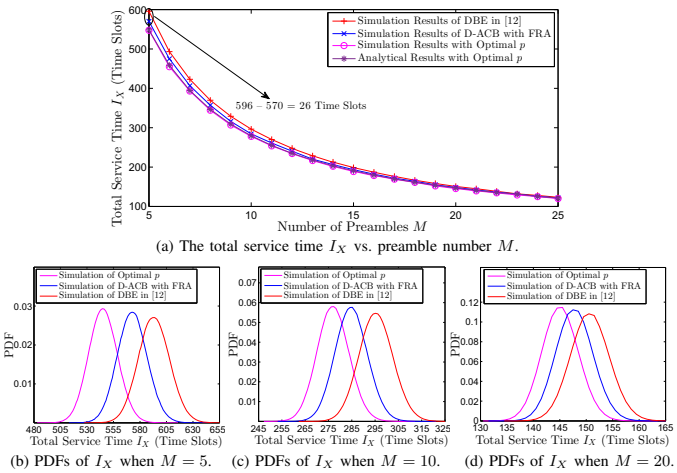
1: Input: N, b.
2: Initialization $i := 0$, $W^{(0)} := 0$, $\widetilde{n}^{(0)} := 0$, $\widehat{p}^{(1)} := 1$, $M_1 := 1$, $\mu_{M_1,p^\star} := M_1(1-2e^{-1})$.
3: **while** cumulative successful transmission $W^{(i)} < N$ **do**
4: Time slot $i := i+1$.
5: Allocate M_i preambles and use ACB factor $\widehat{p}^{(i)}$ in the i^{th} RACH.
6: Monitor the number of successful transmissions: K_i.
7: Monitor the number of preambles that are chosen by more than one user: $c^{(i)}_{M,\widehat{p}}$.
8: Update $W^{(i)} := W^{(i-1)} + K_i$.
9: $\widetilde{p}^{(i)} := \min\left\{1, \widehat{p}^{(i)}\Big(1+\frac{(c^{(i)}_{M_i,\widehat{p}}-\mu_{M_i,p^\star})e}{2M_i}\Big)^{-1}\right\}$, $\widetilde{n}^{(i)} := \frac{M_i}{\widetilde{p}^{(i)}}$.
10: $\widehat{n}^{(i+1)} := \max\big\{0, \widetilde{n}^{(i)} + \max\{-M_i, \widetilde{n}^{(i)} - \widetilde{n}^{(i-1)}\}\big\}$.
11: $M_{i+1} := \max\left\{1, \min\left\{\lceil\frac{\widehat{n}^{(i+1)}}{b}\rceil, 64\right\}\right\}$.
12: $\mu_{M_{i+1},p^\star} := M_{i+1}(1-2e^{-1})$, $\widehat{p}^{(i+1)} := \min\left\{1, \frac{M_{i+1}}{\widehat{n}^{(i+1)}}\right\}$.
13: **end while**

VI. Performance Evaluation

In this section, we present the numerical results of the optimal p scenario and the D-ACB algorithm. In Section VI-A, we present the results on total service time. In Section VI-B, we present results on average number of opportunities consumed by each MTC device.

A. Total Service Time

In this subsection, we present the performance evaluation on the total service time of D-ACB with FRA. Specifically, for a fixed number of random access preambles, we compare the total number of time slots required to serve all MTC devices in the following three schemes: the drift-based backlog estimation (DBE) from [12], our proposed D-ACB with FRA, and the optimal p where the optimal ACB factor is used in each time slot. For the bursty activation model given by the beta distribution in (1), we use the set of parameters $\alpha = 3$ and $\beta = 4$ recommended by 3GPP [21]. The analytical results by using the optimal p are also presented to show the correctness of our analytical model. We vary the number of preambles M from 5 up to 25. The number of users N is equal to 1000. The activation time I_A is 100. We run 1.5 million simulations for each scheme with each value of M. The average simulation results are given in Fig. 3(a). We find that the analytical results and the simulation results by using the optimal p match closely, which validate our analytical model in Section III. Simulation results of D-ACB with FRA and DBE are presented and compared. When the number of preambles allocated to M2M traffic increases, the total service time is reduced as expected. Results show that D-ACB with



(a) The total service time I_X vs. preamble number M.

(b) PDFs of I_X when $M = 5$. (c) PDFs of I_X when $M = 10$. (d) PDFs of I_X when $M = 20$.

Fig. 3. The total service time I_X vs. preamble number M for $N = 1000$ and $I_A = 100$ under beta distribution activation model is presented in (a). The probability density functions (PDFs) of I_X for $M = 5$, $M = 10$, and $M = 20$ are given in (b), (c), and (d), respectively.

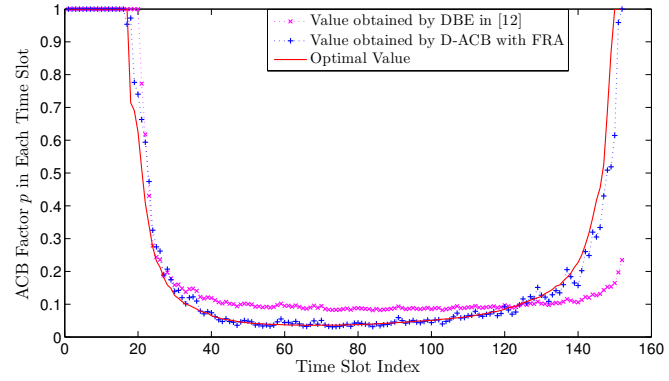
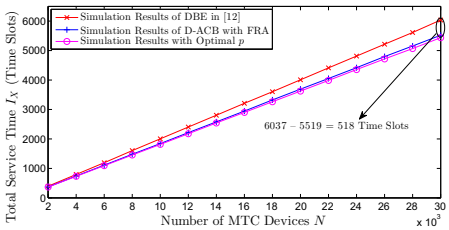


Fig. 4. The dynamic ACB factor p vs. time with $N = 1000$, $I_A = 100$, and $M = 20$ under beta distribution activation model.

FRA and DBE schemes achieve good performance in reducing the total service time, and yet D-ACB with FRA is closer to the optimal performance. D-ACB with FRA outperforms DBE since it requires less expected number of time slots to serve all MTC devices. Specifically, when the number of preambles is 5, D-ACB with FRA saves 26 time slots compared with DBE. In Figs. 3(b)–3(d), we present the probability density functions (PDFs) of I_X for preamble number $M = 5$, $M = 10$, and $M = 20$, respectively. In each figure, three PDFs of I_X by using the scheme of optimal p, our proposed D-ACB with FRA, and the DBE scheme are presented, respectively. We find that the total service time follows the normal distribution. For each scheme with a given M, the mean value of the normal distribution is the average result of I_X obtained under the same simulation setting in Fig. 3(a).

Fig. 4 shows how the ACB factor p in our D-ACB with FRA algorithm (Algorithm 1) varies over time slots in a simulation run. In our algorithm, p is initially set to be 1. The factor is updated in each time slot based on the value in the previous time slot as well as the collision number in the current time slot. Results in Fig. 4 show how the value of p in the proposed algorithm fluctuates around the optimal value. Again, we plot the value of the ACB factor of DBE. We can see from the



(a) The total service time I_X vs. preamble number N.

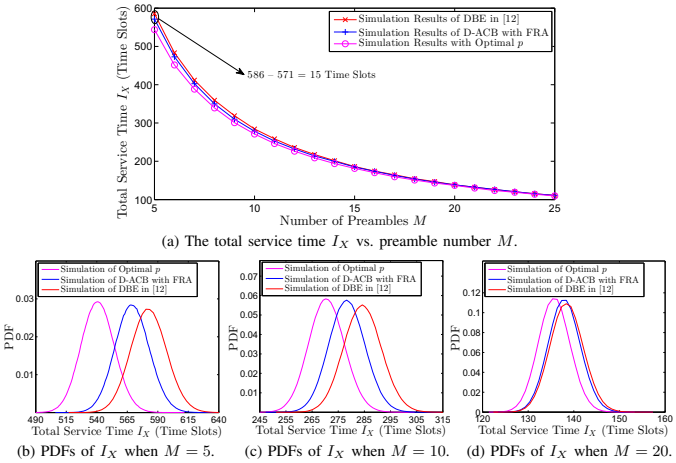
(b) PDFs of I_X when N=10000. (c) PDFs of I_X when N=20000. (d) PDFs of I_X when N=30000.

Fig. 5. The total service time I_X vs. number of MTC devices N for $M = 15$ and $I_A = 100$ under beta distribution activation model is presented in (a). The PDFs of I_X for $N = 10000$, $N = 20000$, and $N = 30000$ are given in (b), (c), and (d), respectively.

figure that our ACB factor stays closer to the optimal value, while the ACB factor of DBE is less accurate. This figure shows why the D-ACB algorithm with FRA can achieve near optimal result and a better performance than DBE in [12].

In practice, the number of MTC devices within a single cell can be very large. We vary the number of devices N from 2000 up to 30000 while the number of preambles is set to be 15 in Fig. 5(a). For each value of N, we run 300000 simulations for each of the following cases: using the DBE in [12], using D-ACB with FRA, and using the optimal p. The average simulation results in Fig. 5(a) show that our Algorithm 1 achieves near optimal performance and performs better than DBE. When the number of MTC devices is 30000, the eNodeB using D-ACB requires 518 fewer time slots to serve all MTC devices than using DBE. This shows the scaling performance of our algorithm. In Figs. 5(b)–5(d), we present the PDFs of I_X for the number of MTC devices $N = 10000$, $N = 20000$, and $N = 30000$, respectively. We find that the total service time I_X by using optimal p, D-ACB with FRA, and DBE follows the normal distribution for each value of N. Specifically, compared with DBE, the PDF curves of our proposed D-ACB with FRA are much closer to the PDF curves of the optimal p due to the accurate ACB factors obtained by Algorithm 1.

Note that the congestion control model is not dependent on the traffic activation model. The same parameters can thus be applied to uniform distribution activation model, *i.e.*, the activations of all the users are uniformly distributed within the activation time. This is also proposed in 3GPP standards [4]. We conduct 1.5 million simulation runs for each simulation setting. The average results are shown in Fig. 6. When the number of preambles M varies from 5 to 25, the proposed algorithm works well under different activation models and outperforms DBE. We also present the PDFs of I_X for $M = 5$, $M = 10$, and $M = 20$ for each scheme. Figs. 6(b)–6(d) show that the total service time I_X with a given simulation setting follows the normal distribution. The mean values of



(a) The total service time I_X vs. preamble number M.

(b) PDFs of I_X when $M = 5$. (c) PDFs of I_X when $M = 10$. (d) PDFs of I_X when $M = 20$.

Fig. 6. The total service time I_X vs. preamble number M for $N = 1000$ and $I_A = 100$ under uniform distribution activation model is presented in (a). The PDFs of I_X for $M = 5$, $M = 10$, and $M = 20$ are given in (b), (c), and (d), respectively.

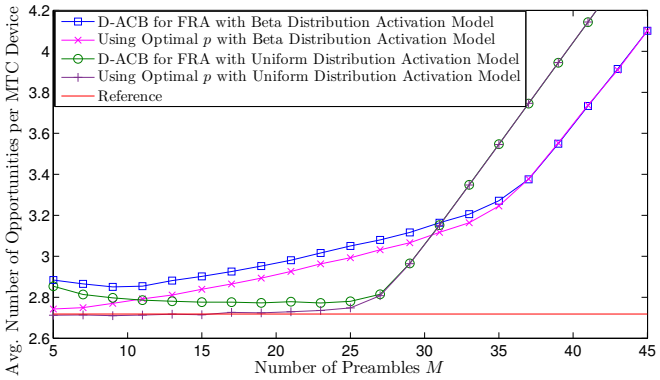


Fig. 7. The average number of random access opportunities per MTC device vs. the number of preambles for $N = 1000$ with beta and uniform distributions.

these normal distributions are the average total service time shown in Fig. 6(a) with the corresponding simulation settings.

B. Random Access Opportunity

In this subsection, we present the simulation results about the random access opportunities consumed by MTC devices. Fig. 7 shows the average number of random access opportunities per MTC device versus the number of preambles by using D-ACB with FRA algorithm (*i.e.*, Algorithm 1) and the optimal p under the beta and uniform distribution activation models, respectively.

The reference line has the value of e, which is the average number of preambles required to successfully serve one MTC device. From the figure, we can observe that the average number of random access opportunities per MTC device increases linearly for large values of M (*i.e.*, $M > 37$ for beta distribution and $M > 27$ for the uniform distribution). Further inspection shows that the slope of these two linear parts is proportional to the value of the activation time I_A. This means that the system is allocating too many preambles for the M2M traffic and most of the initial access requests are handled as soon as the devices are activated (*i.e.*, $I_X \approx I_A$). Since the average number of random access opportunities per MTC

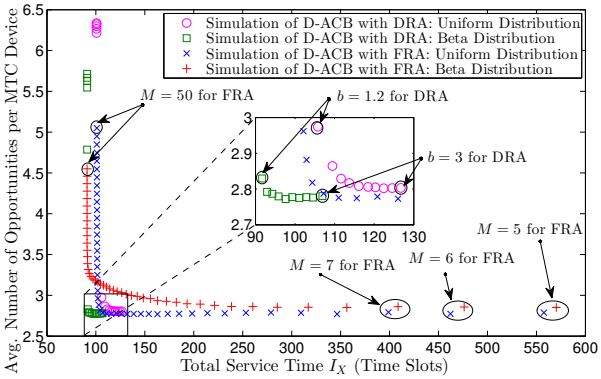


Fig. 8. The average number of random access opportunities per MTC device vs. total service time.

device is given by MI_X/N for D-ACB with FRA, the average number of random access opportunities per MTC device increases linearly after the number of preambles exceeds a threshold. As M increases within the range $5 \leq M \leq 9$ for the beta distribution activation model or within the range $5 \leq M \leq 19$ for the uniform distribution, we observe that the average number of random access opportunities per MTC device in D-ACB with FRA decreases slightly. This is due to the fact that the ACB factor given by Algorithm 1 is less accurate for smaller values of M in this range, which results in a larger value of I_X.

We also plot the average number of random access opportunities per MTC device versus the total service time for D-ACB with DRA and FRA schemes under beta distribution and uniform distribution. For FRA results in Fig. 8, different points are generated using different number of preambles, ranging from 5 to 50. For DRA results in Fig. 8, each data point represents a different parameter b (i.e., $b = 0.2, 0.4, 0.6, \ldots, 3$). The parameter b is introduced in (23) as the scaling factor of the number of backlogged users to dynamically determine the number of preambles for M2M traffic. We chose the activation time $I_A = 100$ and the number of MTC devices $N = 1000$ in the simulation runs.

Note that the average number of random access opportunities per MTC device with FRA is $\frac{MI_X}{N}$, where the number of preambles M is fixed in each simulation run. The average number of opportunities per MTC device with DRA is calculated by $\frac{1}{N}\sum_{i=1}^{I_X} M_i$, where the number of preambles M_i changes for each time slot $i = 1, 2, \ldots, I_X$. The leftmost data points have the minimum total service times, which indicate that all transmissions finish as soon as the devices are activated. As we move to higher service times, fewer resources are consumed per MTC device. As shown in Fig. 8, D-ACB with DRA can achieve both the minimum average number of opportunities per MTC device and the minimum total service time under beta distribution activation model. That is, when the activation time of the MTC devices follows beta distribution, D-ABC with DRA has better performance than D-ACB with FRA in terms of using fewer random access resources while maintaining the minimum total service time. On the other hand, D-ACB with FRA can achieve either the minimum average number of opportunities per MTC device or the minimum total service time, but not both at the same time.

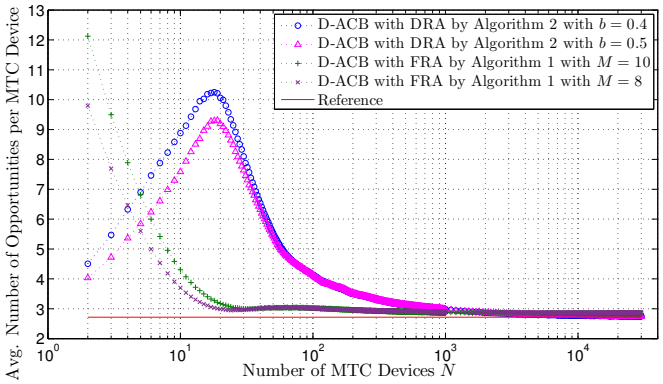


Fig. 9. The average number of opportunities per MTC device vs. the number of MTC devices ($I_A = 2$).

By selecting the value M in FRA appropriately, FRA with uniform distribution activation model may achieve a similar performance as DRA in reducing both the total service time and random access resources. However, determining the proper value of M is not trivial. In contrast, DRA can easily obtain these two objectives by setting $b = 1.2$.

Next, we plot the average number of random access opportunities per MTC device versus the number of MTC devices in FRA and DRA using Algorithms 1 and 2, respectively. For both algorithms, we first increase the number of MTC devices from 2 to 1000 with step size of 1, and then further increase it from 1000 to 30000 with a step size 100. For D-ACB with FRA, where the number of preambles M is fixed, we run simulations with $M = 8$ and $M = 10$, respectively. For D-ACB with DRA, where the number of preambles M_i is dynamically allocated to M2M traffic in each time slot i according to system parameter b, we run simulations with $b = 0.4$ and $b = 0.5$, respectively. The uniform distribution activation model is used with the activation time equal to 2 time slots (*i.e.*, $I_A = 2$). The average results of 1000 simulation runs are shown in Fig. 9. We observe that the average number of random access opportunities per MTC device decreases with the number of MTC devices for D-ACB with FRA (*i.e.*, Algorithm 1). However, for D-ACB with DRA (*i.e.*, Algorithm 2), the number of opportunities per MTC device increases with the number of MTC devices first, and starts to decrease when the number of MTC devices exceeds a threshold. We also find that when the number of MTC devices is large (*i.e.*, $N \geq 5000$), the average number of opportunities per MTC device obtained by both Algorithms 1 and 2 asymptotically approaches constant $e = 2.718$. When the optimal ACB factor is used, each random access opportunity can accommodate e^{-1} successful transmissions on average when the backlog n approaches infinity (Section IV-B). This shows that our algorithms can follow the optimal ACB factor closely.

VII. Conclusion

In this paper, we proposed a congestion control scheme for the bursty traffic scenario of M2M communications in LTE networks. We modeled the system as a multi-channel random access system, and derived the transmission probability matrix. The matrix is used to track how the state vector evolves with time, and to obtain the expected minimum total service time, assuming that the eNodeB is aware of the number of backlogged users in the system. As the random access opportunities are limited resources for LTE networks, we investigated the problem of reducing the average number of random access opportunities per MTC device by dynamic allocating the number of preambles during each time slot based on the number of backlogged users. Then, we considered a more realistic scenario where the eNodeB has no information regarding the number of backlogged users. We proposed an iterative algorithm to adaptively update the ACB factor p, which yields near optimal performance and a reduction in the total service time compared to DBE scheme. As the algorithm is independent of the packet arrival model, we used the same parameters on a different activation models and obtained close-to-optimal performance, which shows the robustness of our algorithm. Simulation results also showed that D-ACB with FRA can achieve as good performance as D-ACB with DRA in reducing the number of random access opportunities at the expense of a longer total service time.

For future work, instead of p-persistent random access model used in our paper, we will explore binary backoff scheme as part of the random access model. We will also study different QoS classes, and set different ACB factors for each access class. More delay-tolerant devices have lower priority in accessing the eNodeB while devices for emergency have higher priority. It is also possible that the entire low priority class is barred to guarantee QoS for high priority class. New activation model can also be considered for different situations.

References

[1] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: From mobile to embedded Internet," *IEEE Comm. Magazine*, vol. 49, no. 4, pp. 36–43, Apr. 2011.

[2] Machina Research Sector Report, "Machine-to-Machine (M2M) communication in consumer electronics 2012-22," Feb. 2013.

[3] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66–74, Apr. 2011.

[4] 3GPP, "Study on RAN improvements for machine-type communications," 3rd Generation Partnership Project (3GPP), TR 37.868 V11.0.0, Oct. 2011.

[5] ——, "Study on enhancements for machine-type communications (MTC)," 3rd Generation Partnership Project (3GPP), TR 22.888 V12.0.0, Mar. 2013.

[6] A. Gotsis, A. Lioumpas, and A. Alexiou, "M2M scheduling over LTE: Challenges and new perspectives," *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 34–39, Sep. 2012.

[7] 3GPP, "Study on machine-type communications (MTC) and other mobile data applications communications enhancements," 3rd Generation Partnership Project (3GPP), TR 23.887 V12.0.0, Dec. 2013.

[8] G. Wang, X. Zhong, S. Mei, and J. Wang, "An adaptive medium access control mechanism for cellular based machine to machine (M2M) communication," in *Proc. of IEEE Int'l Conf. on Wireless Information Technology and Systems (ICWITS)*, Hawaii, HI, Aug. 2010.

[9] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, and K.-C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. on Wireless Communications*, vol. 11, no. 1, pp. 27–32, Jan. 2012.

[10] A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, "Cellular-based machine-to-machine: Overload control," *IEEE Network*, vol. 26, no. 6, pp. 54–60, Nov. 2012.

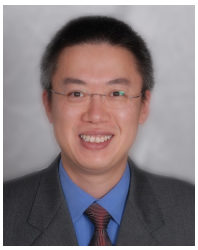
[11] H. Wu, C. Zhu, R. La, X. Liu, and Y. Zhang, "Fast adaptive S-ALOHA scheme for event-driven machine-to-machine communications," in *Proc. of IEEE Vehicular Technology Conf. (VTC-Fall)*, Quebec City, Canada, Sep. 2012.

[12] O. Galinina, A. Turlikov, S. Andreev, and Y. Koucheryavy, "Stabilizing multi-channel slotted aloha for machine-type communications," in *Proc. of IEEE Int'l Symposium on Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.
[13] S.-T. Sheu, C.-H. Chiu, Y.-C. Cheng, and K.-H. Kuo, "Self-adaptive persistent contention scheme for scheduling based machine type communications in LTE system," in *Proc. of Int'l Conf. on Selected Topics in Mobile and Wireless Networking (iCOST)*, Avignon, France, Jul. 2012.
[14] Y. Liu, C. Yuen, J. Chen, and X. Cao, "A scalable hybrid MAC protocol for massive M2M networks," in *Proc. of IEEE Wireless Commun. and Networking Conf. (WCNC)*, Shanghai, China, Apr. 2013.
[15] J.-P. Cheng, C.-H. Lee, and T.-M. Lin, "Prioritized random access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *Proc. of IEEE Globecom Workshop on Machine-to-Machine Communications*, Houston, TX, Dec. 2011.
[16] S.-Y. Lien and K.-C. Chen, "Massive access management for QoS guarantees in 3GPP machine-to-machine communications," *IEEE Communications Letters*, vol. 15, no. 3, pp. 311–313, Mar. 2011.
[17] T. Kwon and J.-W. Choi, "Multi-group random access resource allocation for M2M devices in multicell systems," *IEEE Communications Letters*, vol. 16, no. 6, pp. 834–837, Jun. 2012.
[18] D. Niyato, P. Wang, and D. I. Kim, "Performance modeling and analysis of heterogeneous machine type communications," *IEEE Trans. on Wireless Communications*, vol. 13, no. 5, pp. 2836–2849, May 2014.
[19] P. Osti, P. Lassila, S. Aalto, A. Larmo, and T. Tirronen, "Analysis of PDCCH performance for M2M traffic in LTE," *IEEE Trans. on Vehicular Technology*, vol. 63, no. 9, pp. 4357–4371, Nov. 2014.
[20] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); medium access control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321 V12.7.0, Sep. 2015.
[21] ——, "[70bis#11]-LTE: MTC LTE simulations," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 #71 R2-104663, Aug. 2010.
[22] S. Sesia, I. Toufik, and M. Baker, *LTE – The UMTS Long Term Evolution: From Theory to Practice, 2nd edition*. Wiley, 2011.
[23] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and Its Applications*. CRC Press, 2004.
[24] J. Riordan, *Introduction to Combinatorial Analysis*. John Wiley, 1959.
[25] Z. Wang and V. W. S. Wong, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Trans. on Wireless Communications*, vol. 14, no. 10, pp. 5374–5387, Oct. 2015.
[26] 3GPP, "Evolved universal terrestrial radio access (E-UTRA) physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS 36.211 V12.7.0, Sep. 2015.

Suyang Duan received the B.Eng. degree in Information and Communication Engineering from Zhejiang University, Hangzhou, China, in 2011, and the M.A.Sc. degree in Electrical and Computer Engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2013. He is now with Viavi Solutions Canada. His research interests include machine-to-machine communications and wireless cellular networks.

Vahid Shah-Mansouri (M'12) received the B.Sc. and M.Sc. degrees from the University of Tehran and Sharif University of Technology, Tehran, Iran, in 2003 and 2005, respectively. He obtained his Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2011. From 2011 to 2012, he was a MITACS Postdoctoral Research Fellow at UBC. He joined the School of Electrical and Computer Engineering at the University of Tehran in 2013 where he is currently an Assistant Professor. His research interests include mathematical modeling and optimization of computer and data networks focusing on 5G technology and Internet of Things applications.

Zehua Wang (S'11) received the B.Eng. degree in Software Engineering from Wuhan University, Wuhan, China, in 2009, and the M.Eng. degree in Electrical and Computer Engineering from Memorial University of Newfoundland, St John's, NL, Canada, in 2011. He is currently a Ph.D. candidate at the University of British Columbia (UBC), Vancouver, BC, Canada. His research interests include machine-type communications, device-to-device communications, social networks, and routing and forwarding in mobile ad hoc networks. He has been the recipient of the Four Year Doctoral Fellowship (4YF) at UBC since 2012. He was also awarded the Graduate Support Initiative (GSI) Award from UBC in 2014 and 2015. Mr. Wang served as technical program committee (TPC) members for several conferences including the *IEEE International Conference on Communications (ICC)* 2012, 2014–2016, the *IEEE Global Communications Conference (GLOBECOM)* 2014–2015, and the *IEEE Vehicular Technology Conference (VTC) 2012-Fall, 2016-Fall*.



Vincent W.S. Wong (S'94, M'00, SM'07, F'16) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, and the Internet. Dr. Wong is an Editor of the *IEEE Transactions on Communications*. He is a Guest Editor of *IEEE Journal on Selected Areas in Communications*, special issue on "Emerging Technologies" in 2016. He has served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. He has served as a Technical Program Co-chair of *IEEE SmartGridComm'14*, as well as a Symposium Co-chair of *IEEE SmartGridComm'13* and *IEEE Globecom'13*. He is the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications and the IEEE Vancouver Joint Communications Chapter. Dr. Wong received the 2014 UBC Killam Faculty Research Fellowship.