Dynamic Access Class Barring for M2M Communications in LTE Networks

Suyang Duan*, Vahid Shah-Mansouri† and Vincent W.S. Wong*
Department of Electrical and Computer Engineering
*University of British Columbia, Vancouver, Canada
†University of Tehran, Tehran, Iran
e-mail: *{suyangd, vincentw}@ece.ubc.ca, †vahids@ece.ut.ac.ir

***Abstract*—When incorporating machine-to-machine (M2M) communications into Long Term Evolution (LTE) networks, one of the challenges is the traffic overload due to a large number of machine type communications (MTC) devices with bursty traffic. One approach to tackle this problem is to use the access class barring (ACB) mechanism to regulate the opportunity of MTC devices to transmit request packets. In this paper, we first present an analytical model to determine the expected total access delay of all the MTC devices. For the ideal case that the LTE base station (eNodeB) knows the number of backlogged users, we determine the optimal value of the ACB factor, which can best reduce the congestion and access delay. For the practical scenario, we propose a heuristic algorithm to adaptively change the ACB factor without the knowledge of the number of backlogged users. Results show that the proposed heuristic algorithm achieves near optimal performance in reducing access delay.**

I. Introduction

Machine-to-machine (M2M) communication system is a network which includes a large number of machine type communication (MTC) devices that can communicate with little or no human intervention in order to accomplish specific tasks [1]. According to [2], there will be 12.5 billion MTC devices (excluding smart phones and tablets) in the world in 2020, up from 1.3 billion today.

Using long term evolution (LTE) networks as the air interface for M2M communications has several advantages. The network coverage makes it possible to deploy MTC devices in most urban and rural areas, and the backhaul networks of LTE can provide seamless communication between MTC devices and MTC applications. However, as LTE is optimized for human-to-human (H2H) communications, there are several problems concerning MTC devices accessing LTE networks. One problem is low efficiency, as the actual data packet size for M2M traffic can be much smaller than that of the signalling used in H2H communcations [3]. Another problem is congestion, including air interface congestion and core network (CN) congestion. As described in [4], the number of MTC devices within a cell can be significantly large, *e.g.,* thousands of devices accessing a single base station. If a large number of these devices try to access the base station within a short period of time, congestion, especially signaling congestion, will take place. MTC related signaling congestion and overload can be caused by [5]: a) An external event triggering massive numbers of MTC devices to attach/connect all at once; b) Recurring M2M applications that are synchronized to the exact (half/quarter) hour.

To reduce congestion in an overload condition, several basic solutions are proposed and studied in [4], among which access class barring (ACB) is the main solution. In ACB, the LTE base station, evolved node B (eNodeB), broadcasts a probability p called ACB factor. Each MTC device can access the network with probability p or defer its access by probability $1-p$ for one time slot. Simulation results on fixed ACB factor schemes are presented in [6], [7]. In the literature, there are several papers discussing the congestion problem in M2M communications. In [8], a congestion-aware admission control solution that selectively rejects signaling messages from MTC devices was proposed. The system estimates the probability to reject a random access attempt based on the load of the CN, using a proportional integrative derivative controller. Another congestion control method was discussed in [9], where probability for a packet to transmit is set according to the current traffic load so that the traffic load at the eNodeB is always the optimal value and thus the maximum throughput can be achieved. Using drift analysis, Wu *et al.* in [10] utilized the statistics of consecutive idle and collision slots to reduce access delay under bursty traffic situation. A fast-converging and robust algorithm in estimating the number of backlogged users was proposed. However, both [9] and [10] are limited in that they only considered a single-channel model while in LTE networks, multiple simultaneous transmissions can be accommodated and higher throughput can be achieved.

In this paper, our focus lies in alleviating congestion in radio access network (RAN). We aim to manage random access attempts at the user end to reduce the congestion in an overload condition instead of rejecting access at the eNodeB or the CN. In the case of an emergency, it is crucial that all the information from every single MTC device is collected as soon as possible. Therefore, we need to minimize the total amount of time it takes for all the MTC devices to finish user data transmissions. We consider the use of the ACB scheme with an *adaptive* ACB factor. The contributions of this paper are as follows:

- We first derive an analytical model to determine the minimum time required to handle all the requests from the users where new traffic arrivals follow a beta distribution.
- We propose an algorithm to dynamically adjust the ACB

factor.

- The analytical model is validated by simulation results. Results also show that our proposed heuristic algorithm can achieve near optimal performance.

The rest of the paper is organized as follows: We summarize the random access procedures in LTE networks in Section II. We introduce our analytical model in Section III. In Section IV, we propose an algorithm to adaptively update the ACB factor. Performance evaluation is presented in Section V. The paper is concluded in Section VI.

II. Random Access Procedures in LTE Networks

In this section, we introduce the random access procedures in LTE networks. User data is transmitted through Physical Uplink Shared CHannel (PUSCH) as scheduled transmissions in LTE networks. Asynchronous devices acquire synchronization with the eNodeB and reserve uplink channel using Random Access CHannel (RACH). RACHs are repeated in the system with a certain period. Here the word *channel* refers to the time-frequency resource block that occurs repeatedly, not the physical transmission medium. Each node requiring an uplink channel transmits a preamble in RACH. There are two types of access in RACH: contention-based for regular users and contention-free which provides low-latency service for users with high priority (*e.g.*, handover). We only focus on the contention-based random access here which consists of four steps [11].

In Step 1, each user equipment (UE) randomly selects a sequence called a preamble from a pool known both to UEs and the eNodeB. The transmission of this sequence serves as a request for a dedicated time-frequency resource block in the upcoming scheduling transmission in Step 3. As UEs only transmit the sequence without incorporating their own IDs in the request, when two UEs select the same preamble, the eNodeB will receive the same sequence. In Step 2, the eNodeB acknowledges all the preambles it has successfully received, conveying a timing alignment instruction so that subsequent transmissions can be synchronized. In Step 3, UEs begin using PUSCH to transmit their IDs upon receiving the acknowledgement. If two UEs have selected the same preamble in Step 1, both will be instructed to transmit their IDs within the same time-frequency resource block in Step 3, and then a collision will happen. In Step 4, contention resolution message will be broadcast with the ID of UEs successfully decoded by the eNodeB. If a collision happens while the eNodeB still manages to decode the message in Step 3, it will inform the UE whose Step 3 message is decoded and this successful UE will send an ACK. Unacknowledged UEs will remain silent until the next RACH.

In an LTE cell, 64 preambles are available for random access, among which some are reserved for contention-free access. When MTC devices access LTE networks, they have to share the remaining preambles for contention based access with H2H UEs (*e.g.*, smart phones). In our model, we assume that separate resources are allocated to M2M traffic and H2H traffic. Hence, we only consider how MTC devices compete for dedicated preambles among themselves. Note that random access can only take place within certain time-frequency blocks specified by the eNodeB, *i.e.*, Physical Random Access CHannel (PRACH) which is the physical layer mapping of RACH. For example, when PRACH configuration index is set to 6, RACH will occur every 5 ms within a bandwidth of 180 kHz with a duration ranging from 1 ms to 3 ms [7], [11]. In this paper, we only consider transmissions within the random access channels.

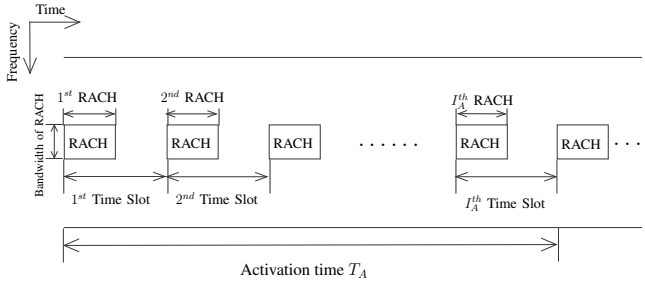

Fig. 1. Random access time slots.

III. System Model

Consider N MTC which have previously registered with an eNodeB. These devices have just recovered from an emergency, *e.g.*, a power black out, and all of them try to re-establish synchronization with the eNodeB. As these devices are not synchronized, they will not be activated all at once, but within a limited time T_A, denoted as the activation time. Each MTC device is activated at time $0 \leq t \leq T_A$ with probability $g(t)$ in which $g(t)$ follows a beta distribution with parameters $\alpha = 3, \beta = 4$ as [7]

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

where $\mathcal{B}(\alpha, \beta)$ is the beta function [12].

Assume there are I_A random access channels within the activation time. The duration of the random access channel is shorter than the interval between two random access channels. We divide the activation time into I_A discrete slots where slot i begins with i^{th} random access channel, as shown in Fig. 1. The length of each slot is equal to the interval between two random access channels. The i^{th} time slot starts at t_{i-1} and ends at 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 random access channel and choose this channel for their first random access attempts. According to [7], the expected number of new activations (arrivals) during each time slot, $\lambda_i, i = 1, 2, \ldots, I_A$, is subject to the distribution of activation traffic $g(t)$ and the total number of devices N as

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

In order to alleviate congestion, the eNodeB broadcasts an ACB factor p as part of the system information before each random access channel. In each random access channel, 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 request packet will be sent. Otherwise, the MTC device stays silent and waits for the next random access channel, in which both the new activations in the next slot and the backlogged users will perform ACB check before transmission. If more than one MTC device selects the same preamble, then a collision will occur at the eNodeB. We assume that when a collision happens, the eNodeB will not be able to decode the collided Step 3 messages, and thus none of the collided MTC devices succeeds in this access channel. Whenever a user fails in one random access channel, it will try to send the sequence during the following channel after ACB check. This scheme uses the deferred first transmission (DFT), where new arrivals are treated as backlogged users.

We are interested in estimating the total time it takes for the eNodeB to collect all users' data. If a preamble is successfully transmitted, the actual user data will then be transmitted without contention on PUSCH via scheduled transmissions and the time it takes is fixed. Therefore, the dominant part is the time for all the MTC devices to successfully transmit Step 1 preamble sequences, which we denote as total service time (TST). In total, it takes the system I_X random access channels before all the requests are successfully transmitted. As I_X is a random variable, we determine its expectation, $\mathbb{E}[I_X]$.

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

When $i > I_A$, no more new activation takes place. The probability that there is no backlogged users at $i \geq I_A$ might be zero. As i increases in the system, $q_{i,0}$ starts growing and approaches 1 eventually. Let $\hat{i}$ denote the smallest $i > I_A$ such that the probability of zero backlogged user in the system is non-zero

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

For $i > \hat{i}$, $q_{i-1,0}$ and $q_{i,0}$ denote the probability that there is no backlogged users in the system at the beginning and at the end of random access channel i, respectively. The probability that the system finishes all transmissions at random access channel i is thus $(q_{i,0} - q_{i-1,0})$. The expectation of TST is then

$$\begin{aligned} \mathbb{E}\left[I_X \mid \hat{i}\right] &= \sum_{i=\hat{i}}^{\infty} i(q_{i,0} - q_{i-1,0}) \\ &= \hat{i}q_{\hat{i},0} + \sum_{i=\hat{i}+1}^{\infty} i(q_{i,0} - q_{i-1,0}). \end{aligned} \tag{4}$$

To derive the expectation of TST, we need to determine how $q_{i,0}$ evolves with time, *i.e.*, as i increases. We consider the evolution of $\mathbf{q_i} = (q_{i,0}, q_{i,1}, \ldots, q_{i,N})$ over time. In total, there are M preambles available in the system. We denote the number of backlogged users before the i^{th} random access opportunity as N_i, the number of users who pass the ACB check and transmit their preamble as $N_i^a,\ N_i^a \leq N_i$, and the number of successful preamble transmissions during that random access channel as K_i. First, we determine the probability of exactly $K_i = k\ (k \leq M)$ successful preamble transmissions when there are $N_i = n$ backlogged users during the current time slot, $\mathbb{P}(K_i = k \mid N_i = n)$. This probability consists of three parts:

1) Among n backlogged users, there are $N_i^a = j$ users who pass the ACB check and transmit their preambles, $\mathbb{P}(N_i^a = j \mid N_i = n)$.
2) Among j transmitted preambles, k preambles succeed.
3) The rest $j - k$ preambles collide.

The first part can be obtained as

$$\mathbb{P}(N_i^a = j \mid N_i = n) = \binom{n}{j} p^j (1-p)^{n-j}. \tag{5}$$

An analog of the second and the third parts would be to place j different objects into M different cells, on the condition that there are exactly k cells that have one object in each of them, and the rest of the cells either have no objects, or at least have 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 in each cell one object, and the number of different combinations is $\binom{j}{k}\binom{M}{k}k!$. Then, we put the remaining $j-k$ objects into $M-k$ different 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 $f(j-k, M-k)$. If $M = k$, then there is no cell to put any objects, so that $f(j-k, 0) = 0$, $j \neq k$. When $j = k$, we have $f(0,0) = 1$. We denote by $S_c,\ c = 1, 2, \ldots, M-k$, the set of events where the c^{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 [13], the cardinality of this set is

$$\begin{aligned} |S| &= |S_1 \cup S_2 \cup \cdots \cup S_{M-k}| \\ &= (-1)^0 \sum_{c=1}^{M-k} |S_c| + (-1)^1 \sum_{c=1}^{M-k} \sum_{l \neq c} |S_c \cap S_l| \\ &\quad + (-1)^2 \sum_{c=1}^{M-k} \sum_{l \neq c} \sum_{\substack{r \neq c \\ r \neq l}} |S_c \cap S_l \cap S_r| \\ &\quad + \cdots + (-1)^{M-k-1} |S_1 \cap S_2 \cap \cdots \cap S_{M-k}|, \end{aligned} \tag{6}$$

in which

$$\sum_{c=1}^{M-k} |S_c| = \binom{M-k}{1}\binom{j-k}{1} 1!(M-k-1)^{j-k-1},$$

$$\sum_{c=1}^{M-k} \sum_{l \neq c} |S_c \cap S_l| = \binom{M-k}{2}\binom{j-k}{2} 2!(M-k-2)^{j-k-2}.$$

If $(M-k) < (j-k)$, then this last term of (6) is $(-1)^{M-k-1}|S_1 \cap S_2 \cap \cdots \cap S_{M-k}|$. Otherwise, when $(j-k) <$

$(M-k)$, the last $M-j$ terms of the series are all zeros, and the last non-zero 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 non-zero term of this series will be

$$\begin{aligned}&(-1)^{u-1}|S_1 \cap S_2 \cap \cdots \cap S_u| \\ &= (-1)^{u-1}\binom{M-k}{u}\binom{j-k}{u}u!(M-k-u)^{j-k-u}. \quad (7)\end{aligned}$$

Therefore,

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

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

$$\begin{aligned}|\overline{S}| &= (M-k)^{j-k} - |S| \\ &= \sum_{c=0}^{u}(-1)^{c}\binom{M-k}{c}\binom{j-k}{c}c!(M-k-c)^{j-k-c} \\ &= f(j-k, M-k). \quad (8)\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathbb{P}(K_i = k \mid N_i = n) \\ &= \sum_{j=0}^{n} Pr(N_i^a = j \mid N_i = n)\frac{\binom{j}{k}\binom{M}{k}k!f(j-k, M-k)}{M^j} \\ &= \sum_{j=0}^{n}\binom{n}{j}p^j(1-p)^{n-j}\binom{j}{k}\binom{M}{k}k! \\ &\quad \times \frac{\sum_{c=0}^{u}(-1)^c\binom{M-k}{c}\binom{j-k}{c}c!(M-k-c)^{j-k-c}}{M^j}. \quad (9)\end{aligned}$$

We introduce an $(N+1)\times(N+1)$ transmission probability matrix, $\mathbf{R}$, where the element r_{st}, $s = 0, 1, \ldots, N, t = 0, 1, \ldots, N$, is equal to $\mathbb{P}(K_i = s - t \mid N_i = s)$, which is the probability that given s backlogged users in the system, $s-t$ users pass the ACB check and transmit successfully without collision. Note that $r_{0,0} = 1$.

For time slot $i > I_A$, there is no new activation in the system. In this case, matrix $\mathbf{R}$ can relate vectors $\mathbf{q_{i+1}}$ and $\mathbf{q_i}$ as $\mathbf{q_{i+1}} = \mathbf{R}\mathbf{q_i}$. For time slot $i = 1, \ldots, I_A$, we need to take into account the new arrivals when relating $\mathbf{q_{i+1}}$ and $\mathbf{q_i}$. Assume that z MTC devices have been activated until the beginning of time slot $(i+1)$, we have $q_{i,n} = 0$ for $z < n \leq N$

$$\mathbf{q_i} = (q_{i,0}, q_{i,1}, q_{i,2}, \ldots, q_{i,z}, 0, 0, \ldots, 0).$$

The vector $\mathbf{q_i}$ shows the probability for the number of backlogged users after completion of the i^{th} random access channel. If the number of newly activated devices in time slot $(i+1)$ is λ_{i+1}, we define $\mathbf{q'_{i+1}}$ by shifting $\mathbf{q_i}$ λ_{i+1} units as

$$\mathbf{q'_{i+1}} = (\underbrace{0, 0, \ldots, 0}_{\lambda_{i+1}}, q_{i,1}, q_{i,2}, \ldots, q_{i,z}, 0, 0, \ldots, 0).$$

The vector $\mathbf{q'_{i+1}}$ represents the probability for the number of backlogged users right before the start of random access channel $(i+1)$. Therefore, we can obtain $\mathbf{q_{i+1}}$ by $\mathbf{q_{i+1}} = \mathbf{R}\mathbf{q'_{i+1}}$. Using these equations, the state vector of each time slot can be derived starting from $i = 1$. Consequently, using (4), TST can thus be estimated.

The ACB factor p plays an important role in the performance of congestion control in a random access channel. Therefore, it is of interest to find the optimal p. If $N_i^a = j$ users among $N_i = n$ backlogged ones pass the ACB check, each of them will choose from M preambles with equal probability, $\frac{1}{M}$. Consider preamble m and let $D_m = 0, 1, c$ respectively denote the cases where the preamble m is selected by none of the users, by exactly one user, and by more than one user for $m = 1, 2, \ldots, M$. The probability that only one user selects preamble m is

$$\mathbb{P}(D_m = 1 \mid N_i^a = j) = \binom{j}{1}\frac{1}{M}\Big(1 - \frac{1}{M}\Big)^{j-1}. \quad (10)$$

As each preamble is independent of others, the expected number of successful transmissions is

$$\begin{aligned}\mathbb{E}\left[K_i \mid N_i^a = j\right] &= \sum_{m=1}^{M}\mathbb{P}(D_m = 1 \mid N_i^a = j) \\ &= M\binom{j}{1}\frac{1}{M}\Big(1 - \frac{1}{M}\Big)^{j-1}. \quad (11)\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left[K_i \mid N_i = n\right] \\ &= \sum_{j=1}^{n}\mathbb{P}(N_i^a = j \mid N_i = n)M\binom{j}{1}\frac{1}{M}\Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= \sum_{j=1}^{n}\binom{n}{j}p^j(1-p)^{n-j}\binom{j}{1}\Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= np\Big(1 - \frac{p}{M}\Big)^{n-1}. \quad (12)\end{aligned}$$

The minimum TST can be achieved when the number of successful transmissions during each time slot is maximized. In other words, the maximum system throughput corresponds to the minimum TST. By taking the derivative of (12) with respect to p, we have

$$\frac{\mathrm{d}}{\mathrm{d}p}\mathbb{E}(K_i \mid N_i = n) = n\Big(1 - \tfrac{p}{M}\Big)^{n-2}\Big(1 - \tfrac{np}{M}\Big). \quad (13)$$

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

$$p^* = \min\left(1, \tfrac{M}{n}\right). \quad (14)$$

IV. A Heuristic Algorithm to Update p

In this section, we present a heuristic algorithm to adaptively update the ACB factor p. In a real system, the eNodeB cannot acquire the number of backlogged users in the system. The information it has is limited to the number of successful

transmissions and the number of collisions during each time slot, as well as the total number of M2M devices that have registered in the system, N. There is an inherent trade-off in choosing the ACB factor p. When p is too large, there will be a lot of preambles transmitted in the air, and there will be collisions on most of the preambles. On the other hand, when p is too small, very few users will be able to pass ACB check and transmit their preambles, resulting in fewer collisions but under-utilization of network resources.

As users select with equal probability among all M preambles, the probability that preamble m is selected by a user in a time slot is p/M. The probability that no user chooses a certain preamble m is

$$\mathbb{P}(D_m = 0 \mid N_i = n) = \left(1 - \frac{p}{M}\right)^n. \tag{15}$$

We can also obtain the probability that one preamble is selected by exactly one user as

$$\mathbb{P}(D_m = 1 \mid N_i = n) = \binom{n}{1} \frac{p}{M} \left(1 - \frac{p}{M}\right)^{n-1}. \tag{16}$$

Therefore, the probability of collision $\mathbb{P}(D_m = c \mid N_i = n) = 1 - \mathbb{P}(D_m = 0 \mid N_i = n) - \mathbb{P}(D_m = 1 \mid N_i = n)$. The expected number of preambles with collision, $\mathbb{E}[C]$, is thus

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

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

$$\mathbb{E}[C] = M\left(1 - \left(1 - \tfrac{1}{n}\right)^n - \left(1 - \tfrac{1}{n}\right)^{n-1}\right). \tag{17}$$

For large values of n, $\mathbb{E}[C]$ goes to $M(1 - 2e^{-1})$. We denote this average value as C_0. We monitor the level of collisions as a factor to adjust p. We compare the collision level of the previous 3 time slots to C_0 as a reference to estimate the current backlogged situation. The number of collisions larger than C_0 suggests an overload situation while smaller than C_0 means under-utilization.

Our heuristic algorithm to adjust parameter p is shown in Algorithm 1. In this algorithm, μ_1, $\nu_2 > 1$, ν_1, $\mu_2 < 1$ are design parameters used to adaptively adjust p. They are obtained via simulations. We use W_i to denote the cumulative number of successful transmissions up to time slot i and $\widehat{C}$ to denote the the average number of collisions during the last three time slots. At time slot i, if $W_i \leq 0.5N$, p is updated by comparing the $\widehat{C}$ and C_0. When $W_i > 0.5N$, *i.e.*, half of all the preambles have been successfully transmitted, we assume that all the devices have been activated and are currently in the backlogged status. Then p is set to the optimal value, which can be derived based on this assumption (Step 17).

At time slot i, the eNodeB updates W_i as $W_i = W_{i-1} + K_i$, where K_i is the number of users successfully transmitting request packets in slot i (Step 6). If W_i is less than $0.5N$, the eNodeB updates $\widehat{C}$ using the observed levels of collisions

Algorithm 1 Algorithm for Adaptively Updating p

1: input: C_0, N, design parameters μ_1, ν_2, ν_1, μ_2
2: set $i := 0$, $p := 1$, $W_0 := 0$, $\widehat{C} := C_0$
3: **while** cumulative successful transmission $W_i < N$ **do**
4: time slot $i := i + 1$
5: monitor C_i, K_i
6: update $W_i := W_{i-1} + K_i$
7: **if** $W_i \leq 0.5N$ **then**
8: **if** $i > 3$ **then**
9: update $\widehat{C} := \frac{1}{3}(C_{i-1} + C_{i-2} + C_{i-3})$
10: **end if**
11: **if** $\widehat{C} \geq \mu_1 C_0$ **then**
12: $p := \nu_1 p$
13: **else if** $\widehat{C} \leq \mu_2 C_0$ **then**
14: $p := \nu_2 p$
15: **end if**
16: **else**
17: $p := \min\left(\frac{M}{N - W_i}, 1\right)$
18: **end if**
19: **end while**

(Step 9). For the first three time slots, we have $\widehat{C} = C_0$. If the average level of collisions is higher than a threshold $\mu_1 C_0$, this means the system is overloaded and we decrease p by coefficient μ_1 (Step 12). If the average number of collisions is below a threshold $\mu_2 C_0$, the ACB factor p is increased by coefficient μ_2 (Step 14). The ACB factor does not change if $\widehat{C}$ is between these thresholds (*i.e.*, $\mu_2 C_0 < \widehat{C} < \mu_1 C_0$).

V. Numerical Results

In this section, we present the numerical results of the above analysis. We adopt parameters from [6], [7] to make our model more practical. The activation time $I_A = 100$. First we have a fixed number of users, $N = 1000$, and vary the number of preambles, M. Then we fix the number of preambles M to be 15, and vary the number of users, N. Based on simulation results, we heuristically set the parameters of our algorithm as $\mu_1 = 1.5, \mu_2 = 0.6, \nu_1 = 0.5, \nu_2 = 1.5$ in all our simulations. We also include the fixed ACB factor scenario as a reference where $p = \frac{M}{N}$.

Fig. 2 shows TST against the number of preambles M, which increases from 5 up to 50. It can be seen that TST decreases with increasing number of preambles. For the optimal p scenario, analytical results and simulation results match, which validate our analysis. As we can see, our algorithm is good in reducing TST, which has close to optimal performance and is much better than the fixed p scenario.

Fig. 3 shows how the ACB factor p in our proposed algorithm is being updated over time. As our algorithm is a two-step approach, the value of p changes in different ways for each step. In the first step when fewer than half of all the request packets have been transmitted, the value of p fluctuates around the optimal value p^* as Step 7 to Step 15 in Algorithm 1 suggest. When more than half have been transmitted, we assume that all devices have been activated and are in the

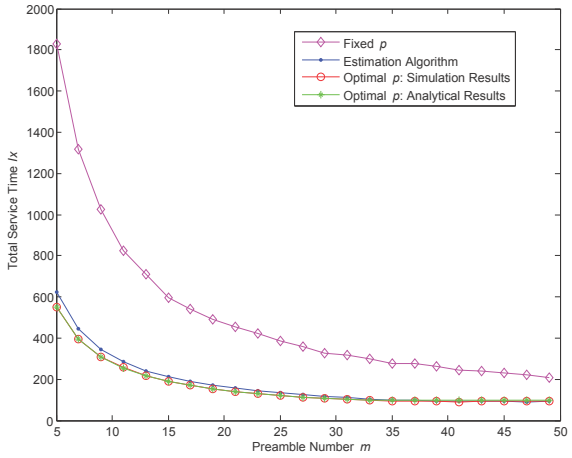

Fig. 2. TST vs preamble number M with $N = 1000$ and $I_A = 100$.

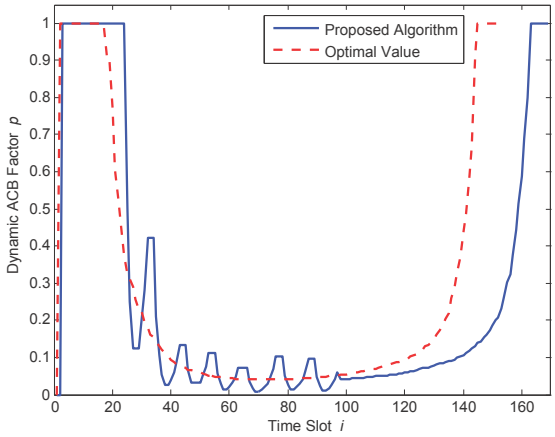

Fig. 3. The dynamic ACB factor p vs number of time slots with $N = 1000$, $I_A = 100$, $M = 20$.

backlogged state. This approximately corresponds to $i = 100$ in Fig. 3. Then, the value of p is adjusted based on the number of remaining backlogged users in the system as in Steps 16 to 18, which gradually increases as more and more packets are successfully transmitted until all the devices finish transmitting preambles.

In reality, the number of M2M devices within a single cell could be significantly large. We vary the number of devices from 1000 up to 30000 in Fig. 4. Results show that our estimation can still achieve near optimal performance. Compared to the fixed ACB scenario, our algorithm yields much better performance in reducing TST. This shows the scaling behavior of our algorithm.

VI. Conclusion

In this paper, we considered an overloaded M2M communication system. We presented how ACB factor can be dynamically updated to reduce TST under such circumstances. We started with the analytical model of an optimal case where the eNodeB knows the number of backlogged users. Then, we proposed a heuristic model where the eNodeB updates the ACB factor adaptively based on the number of collisions in previous time slots. Simulation results showed that our scheme

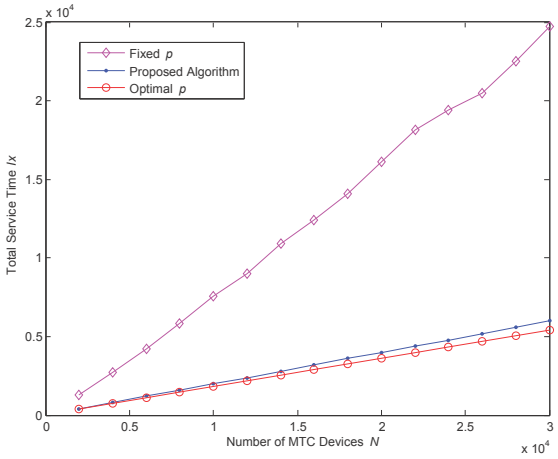

Fig. 4. TST vs number of MTC devices N with $M = 15$, $I_A = 100$.

can achieve near optimal performance compared to the optimal case, and can greatly reduce TST compared to the scenario of fixed ACB factor. For future work, different QoS classes can be introduced, each with separate ACB factors to meet different QoS requirements. Backoff can also be considered instead of the p-persistent model used in the paper.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Dr. Hu Jin for his valuable comments and viewpoints that have helped to shape this paper.

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] 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.

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

[5] ——, "System improvements for machine-type communications," 3rd Generation Partnership Project (3GPP), TR 23.888 V11.0.0, Sep. 2012.

[6] ——, "MTC simulation results with specific solutions," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 #71 R2-104662, Aug. 2010.

[7] ——, "[70bis#11]-LTE: MTC LTE simulations," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 #71 R2-104663, Aug. 2010.

[8] 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.

[9] 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.

[10] 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 VTC-Fall*, Quebec City, Canada, Sep. 2012.

[11] S. Sesia, I. Toufik, and M. Baker, *LTE–The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2009.

[12] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and Its Applications*. CRC Press, 2004.

[13] J. Riordan, *Introduction to Combinatorial Analysis*. John Wiley, 1959.