Recursive Access Class Barring for Machine Type Communications with PUSCH Resource Constraints

Han Seung Jang[†], Hu Jin[‡], Bang Chul Jung[§], and Tony Q.S. Quek[¶]

[†] School of Electrical, Electrnoic Communication, and Compter Engineering, Chonnam National University, Yeosu, South Korea
[‡] Division of Electrical Engineering, Hanyang University, Ansan, South Korea
[§] Department of Electronics Engineering, Chungnam National University, Daejeon, South Korea
[¶]Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singpapore
Email: hsjang@jnu.ac.kr, hjin@hanyang.ac.kr, bcjung@cnu.ac.kr, tonyquek@sutd.edu.sg

***Abstract*—In massive cellular internet-of-things (IoT) networks, periodic and sporadic traffic may be well processed, while bursty traffic may cause an unexpected network congestion or overload problem. Thus, in this paper, we propose a generalized random access (RA) control mechanism considering whole steps of RA procedure and available resources at each step for handling bursty traffic in cellular IoT networks. The proposed RA control mechanism mainly consists of the estimation method for the number of backlogged nodes and the computation method for access class barring (ACB) factors. Through extensive computer simulations, the proposed RA control mechanism shows the enhanced performance in terms of the total service time, the average access delay, and the energy efficiency, compared to the conventional RA control mechanism, which only focuses on controlling preamble transmissions at the first step of the RA procedure.**

***Index Terms*—Machine type communications, internet-of-things, random access, access class barring, preamble, PUSCH**

I. Introduction

In emerging cellular internet-of-things (IoT) and machine-to-machine (M2M) communication networks, a massive number of IoT devices generate various types of data traffic [1]. Especially, periodic and sporadic traffic may be well processed in the typical cellular networks such as LTE and LTE-advanced, while bursty traffic may cause an unexpected network congestion/overload and resource shortage problems [2]. Initial random access (RA) procedure may be the first bottleneck since a huge number of idle nodes wake up and attempt RAs simultaneously for switching to the radio resource control (RRC) connected mode [3]. Hence, various techniques have been proposed to resolve the RA overload problem in the cellular IoT systems [4].

The access class barring (ACB) scheme has been known as a leading technology in push-based access control schemes, which is very effective to control bursty traffic in cellular IoT networks [5]. In general, the ACB schemes inherently control the number of simultaneous RA attempts with an ACB factor ranging between 0 and 1 [6], [7]. In particular, Jin *et al.* [7] proposed the recursive Pseudo-Bayesian ACB method for bursty traffic, which dynamically updates the backlog estimation based on the number of available preambles and the number of unused preambles, and controls the number of simultaneous RA attempts by ACB factors. Even though the backlog estimation is quite well matched with the actual number of backlogged devices, it fragmentarily considers the first step of the RA procedure associated with available preambles in order to control bursty RA traffic.

Likewise, ACB schemes in literature have little or not focused on available random access response (RAR) messages at the second step of the RA procedure and available physical uplink shared channel (PUSCH) resources at third step of the RA procedure, but available preambles at the first step of the RA procedure. Even though a larger number of nodes successfully transmit their preambles without collisions at the first step of the RA procedure, there may be a much severer bottleneck at the second and third steps of the RA procedure due to lack of RAR messages and PUSCH resources [8]–[10].

Hence, in this paper, we propose a generalized RA control mechanism considering whole steps of the RA procedure and available resources at each step for handling bursty traffic in cellular IoT networks. The proposed RA control mechanism mainly consists of the computation method for ACB factors and the estimation method for the number of backlogged nodes. Through extensive computer simulations, we compare the performance of the proposed RA control mechanism with that of the conventional RA control mechanism [7], which only focuses on controlling preamble transmissions at the first step of the RA procedure, in terms of the total service time, the average RA delay, and the energy efficiency.

II. System Model

In the proposed RA control mechanism, we will consider following parameters:

- M: the number of available preambles at step 1.
- Q: the number of available RAR messages at step 2.
- U: the number of available PUSCH resources at step 3.
- K: the number of allocable PUSCH resources at step 2.
- ν_i: the est. number of backlogged nodes for the ith slot.
- p_i: ACB factor for the ith slot.

In particular, a single RAR message can convey a single uplink grant for PUSCH resource [11], the number of allocable

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

PUSCH resources at step 2 is obtained as

$$K = \min\{Q, U\}. \tag{1}$$

Thus, we only consider the number of allocable PUSCH resources K instead of Q and U throughput the paper.

A. Overall Steps

The overall steps include the access check (*Step 0*) and the four-step contention-based RA procedure (*Step 1-4*) [12]:

(***Step 0***) **Access check**: The ACB factor of the ith PRACH $p_i \in [0, 1]$ is notified by the eNodeB before the beginning of the ith PRACH slot. Then, each of active nodes generates a random number $q \in [0, 1]$ and compares q with p_i. If $q \leq p_i$, the active node attempts an RA on the ith PRACH. Otherwise, it defers an RA attempt to the $(i + 1)$th PRACH.

(***Step 1***) **Preamble transmission and detection**: Each of active nodes with $q \leq p_i$ randomly selects a single preamble out of M available preambles, and then transmits the selected preamble on the ith PRACH slot. If two or more nodes select the identical preamble, a preamble collision occurs. However, during the preamble detection, the eNodeB can only identify whether a specific preamble is detected or not, but it cannot recognize that the detected preamble is transmitted by a single node or multiple nodes.

(***Step 2***) **Random access response**: The eNodeB sends RAR messages, each of which conveys the identity of detected preamble, timing advance information, and an uplink resource grant for RA-step 3. If the number of allocable PUSCH resources K is smaller than the number of detected preambles, it randomly chooses K detected preambles, and gives PUSCH resource grants only to them via RAR messages.

(***Step 3***) **Uplink data transmission**: Using the PUSCH resource informed by the RAR message, the corresponding node transmits the uplink data. If two or more nodes use the identical preamble and the corresponding RAR message, all relevant nodes transmit their own data on the same PUSCH resource, which makes the eNodeB fails to decode the data. However, from this failed data decoding, a preamble collision can be recognized.

(***Step 4***) **Contention resolution & ACK**: If the eNodeB successfully decodes the data transmitted by a single node, it sends back an ACK message including the node ID obtained from the decoded data, otherwise the eNodeB sends nothing back in order to notify a preamble collision. The nodes which do not obtain their IDs regard the situation as a preamble collision, and they reattempt RA from the step 0.

B. Bursty Traffic Model

Each of N machine nodes is assumed to activate at the time $x \in (0, T_{\text{act}})$, and the activation time x follows Beta distribution with parameters $\alpha = 3$ and $\beta = 4$, i.e., [13]

$$f_X(x) = \frac{x^{\alpha-1}(T_{\text{act}} - x)^{\beta-1}}{(T_{\text{act}})^{\alpha+\beta-1}B(\alpha, \beta)},$$

where $B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1 - x)^{\beta-1}dx$ represents the beta function. Within a bursty time period T_{act}, I_{act} slots exist, and the expected number of newly activated nodes in the ith RA slot is given by $\lambda_i = N \int_{t_{i-1}}^{t_i} f_X(x)dx$ for $i = 1, 2, \ldots, I_{\text{act}}$.

III. Proposed Random Access Control Mechanism Using the Optimal ACB factors

We propose an RA control mechanism, which consists of two main parts: the computation algorithm for the optimal ACB factor and the estimation/update algorithm for the number of backlogged nodes. In this Section, we first derive the average numbers of detected, collision-free, and collided preambles at the first step of the RA procedure, and then, we calculate the average access throughput considering available preambles and allocable PUSCH resources. In the last part of the Section, we propose a computation algorithm for the optimal ACB factor in terms of maximizing the average access throughput. In the following Section, the estimation/update algorithm for the number of backlogged nodes will be discussed.

A. Average numbers of detected, collision-free, and collided preambles at the first step of the RA procedure

At the first step of the RA procedure, each of active nodes which passed the access check, transmits an arbitrary preamble out of M available preambles on PRACH slot, and then, the eNodeB detects preambles. In general, detected preambles is classified into collision-free preambles (transmitted by a single node) and collided preamble (transmitted by two or more nodes), but the eNodeB cannot identify the detected preamble is collision-free or collided at this time. Let first define the following discrete random variables:

- D: the number of detected preambles,
- S: the number of collision-free preambles,
- C: the number of collided preambles,

where $D = S + C$. Let E_i denote the event that the ith preamble is detected, and the probability of the event E_i with m RA-attempting nodes is written as $\Pr\{E_i|m\}$. The joint probability that any d preambles out of M preambles are detected, and, each of any s preambles out of d detected preambles is collision-free with m RA-attempting nodes can be written as

$$\begin{aligned}
\Psi_d^M(s|m) &= \binom{M}{d}\binom{d}{s}\binom{m}{s}s!\frac{(d-s)^{m-s}}{M^m} \\
&= \binom{M}{M-d+s}\binom{M-d+s}{s}\binom{m}{s}s!\frac{(d-s)^{m-s}}{M^m}.
\end{aligned} \tag{2}$$

In Eq. (2), $\binom{M}{d}\binom{d}{s}$ represents the total number of cases that any d preambles are detected out of M preambles, and each of any s preambles out of d detected preambles is collision-free. $\binom{m}{s}s!(d-s)^{m-s}$ represents the total number of cases that each of s nodes among m RA-attempting nodes transmits an exclusive preamble, and each of remaining $(m - s)$ nodes chooses one of preambles among $(d - s)$ preambles. Then, $\binom{M}{d}\binom{d}{s}\binom{m}{s}s!(d-s)^{m-s}$ is divided by M^m representing the total number of cases that each of m RA-attempting nodes selects one of preambles among M preambles.

Based on the inclusion-exclusion principle, $\Pr\{E_1 \cup \cdots \cup E_L|m\}$ can be written by

$$\Pr\left\{\cup_{i=1}^{L} E_i|m\right\} = \sum_{i=1}^{L}(-1)^{i+1}\sum_{j=0}^{i}\Psi_i^L(j|m)$$
$$= \sum_{i=1}^{L}\sum_{j=0}^{i}(-1)^{i+1}\binom{L}{i}\binom{i}{j}\binom{m}{j}j!\frac{(L-i)^{m-j}}{L^m}, \quad (3)$$

where $\Pr\left\{\cup_{i=1}^{L} E_i|m\right\}$ denotes the probability that at least one preamble among L preambles is detected with m RA-attempting nodes. The complementary probability to $\Pr\left\{\cup_{i=1}^{L} E_i|m\right\}$ is written as

$$\Phi(L|m) = \Pr\left\{\cap_{i=1}^{L} E_i|m\right\} = 1 - \Pr\left\{\cup_{i=1}^{L} E_i|m\right\}$$
$$= 1 - \sum_{i=1}^{L}\sum_{j=0}^{i}(-1)^{i+1}\binom{L}{i}\binom{i}{j}\binom{m}{j}j!\frac{(L-i)^{m-j}}{L^m}$$
$$= \sum_{i=0}^{L}\sum_{j=0}^{i}(-1)^{i}\binom{L}{i}\binom{i}{j}\binom{m}{j}j!\frac{(L-i)^{m-j}}{L^m}, \quad (4)$$

which represents the probability that all L preambles are detected with m RA-attempting nodes. Using $\Psi_d^M(s|m)$ and $\Phi(L|m)$, we obtain

$$\Pr\{D=d, S=s|m\} = \Psi_d^M(s|m)\Phi(d-s|m-s)$$
$$= \sum_{i=0}^{d-s}\sum_{j=0}^{i}\binom{M}{d}\binom{d}{s}\binom{m}{s}s!\frac{(d-s)^{m-s}}{M^m}\times$$
$$(-1)^i\binom{d-s}{i}\binom{i}{j}\binom{m-s}{j}j!\frac{(d-s-i)^{m-s-j}}{(d-s)^{m-s}}. \quad (5)$$

Now, we consider following joint probability that s preambles out of d detected preambles are collision-free when m of n backlogged nodes simultaneously attempt RAs:

$$\Pr\{d,s,n,m|p,\nu\} = \Pr\{d,s|m\}\underbrace{\Pr\{m|p,n\}}_{\mathbb{B}_m^n(p)}\underbrace{\Pr\{n|\nu\}}_{\mathbb{P}(n|\nu)}, \quad (6)$$

where we assume that $\mathbb{P}(n|\nu) = \frac{\nu^n}{n!}e^{-\nu}$ follows the Poisson distribution with mean of ν, and $\mathbb{B}_m^n(p) = \binom{n}{m}p^m(1-p)^{(n-m)}$ represents Binomial distribution with ACB factor of p. By summing up with regard to n and m, we have the joint probability that s preambles out of d detected preambles are collision-free with given p and ν

$$\Pr\{D=d, S=s|p,\nu\} = \sum_{n=0}^{\infty}\sum_{m=0}^{n}\Pr\{d,s,n,m|p,\nu\}$$
$$= \binom{M}{d}\binom{d}{s}e^{-p\nu}\left(\frac{p\nu}{M}\right)^s\left(-1-\frac{p\nu}{M}+e^{\frac{p\nu}{M}}\right)^{d-s}. \quad (7)$$

We omit the detailed derivation of (7) due to space limit. From (7), we obtain $\Pr\{D=d|p,\nu\}$ and $\Pr\{S=s|p,\nu\}$ as follows:

$$\Pr\{D=d|p,\nu\} = \sum_{s=0}^{d}\Pr\{D=d,S=s|p,\nu\}$$
$$= \binom{M}{d}e^{-p\nu}\left(-1+e^{\frac{p\nu}{M}}\right)^d. \quad (8)$$
$$\Pr\{S=s|p,\nu\} = \sum_{d=0}^{M}\Pr\{D=d,S=s|p,\nu\}$$
$$= \binom{M}{s}e^{-p\nu}\left(\frac{p\nu}{M}\right)^s\left(-\frac{p\nu}{M}+e^{\frac{p\nu}{M}}\right)^{M-s}. \quad (9)$$

In addition, the number of collided preambles C is easily obtained by $C = D - S$, and thus, we have $\Pr\{C=c, S=s|p,\nu\}$ by substituting $s+c$ for d in Eq. (7). As a result, $\Pr\{C=c|p,\nu\}$ is obtained by,

$$\Pr\{C=c|p,\nu\} = \sum_{s=0}^{M-c}\Pr\{C=c,S=s|p,\nu\}$$
$$= \binom{M}{c}e^{-p\nu}\left(-1-\frac{p\nu}{M}+e^{\frac{p\nu}{M}}\right)^c\left(1+\frac{p\nu}{M}\right)^{M-c}. \quad (10)$$

Next, the expected values of D, S and C are calculated with given p and ν, respectively, by

$$\mathbb{E}[D|p,\nu] = \sum_{d=0}^{M}d\binom{M}{d}e^{-p\nu}\left(-1+e^{\frac{p\nu}{M}}\right)^d$$
$$= M\left(1-e^{-\frac{p\nu}{M}}\right). \quad (11)$$
$$\mathbb{E}[S|p,\nu] = \sum_{s=0}^{M}s\binom{M}{s}\left(\frac{p\nu}{M}\right)^s e^{-p\nu}\left(-\frac{p\nu}{M}+e^{\frac{p\nu}{M}}\right)^{M-s}$$
$$= p\nu e^{-\frac{p\nu}{M}}. \quad (12)$$
$$\mathbb{E}[C|p,\nu] = M\left(-1-\frac{p\nu}{M}+e^{\frac{p\nu}{M}}\right)e^{-\frac{p\nu}{M}}. \quad (13)$$

B. Average Access Throughput and Optimal ACB factor in terms of average access throughput maximization

After detecting preambles, the eNodeB allocates PUSCH resources to detected preambles via RAR messages. When the allocable PUSCH resources are insufficient for the detected preambles, some nodes cannot receive PUSCH resource grants at the second step of the RA procedure. Hence, the successful access is achieved only in case that a node transmits an exclusive preamble and obtain a PUSCH resource grant. Let T denote the access throughput, i.e, the number of nodes that succeed in random access. For the calculation of the access throughput, we need to consider two cases: $D \leq K$ and $D > K$. Although $\Pr\{T=t|p,\nu\}$ depend on p and ν, we suppress them for notational simplicity. Thus, we can write $\Pr\{T=t\}$ as follows

$$\Pr\{T=t\} = \Pr\{T=t, D\leq K\} + \Pr\{T=t, D>K\}$$
$$= \Pr\{D\leq K, S=t\} + \Pr\{D>K, S\geq t\}\mathbb{B}_t^s\left(\tfrac{K}{d}\right), \quad (14)$$

where $\mathbb{B}_t^s\left(\frac{K}{d}\right) = \binom{s}{t}\left(\frac{K}{d}\right)^t\left(1-\frac{K}{d}\right)^{s-t}$. Then, the expected value of T is calculated by

$$\mathbb{E}[T|p,\nu] = \sum_{t=0}^{K}\sum_{d=0}^{K} t\Pr\{D=d, S=t\}+ \sum_{t=0}^{K}\sum_{d=K+1}^{M}\sum_{s=t}^{d} t\Pr\{D=d, S=s\}\mathbb{B}_t^s\left(\tfrac{K}{d}\right). \quad (15)$$

We should find the optimal p to maximize the average access throughput by setting the following optimization problem:

$$p^* = \underset{0\le p\le 1}{\arg\max}\ \mathbb{E}[T|p,\nu] \quad (16)$$

Especially, when $K = M$, $\mathbb{E}[T|p,\nu] = \mathbb{E}[S|p,\nu]$, thus, the optimal p^* is found by taking the first derivative of $\mathbb{E}[S|p,\nu] = p\nu e^{-\frac{p\nu}{M}}$ with respect to p and letting it to be equal to 0, i.e,

$$p^* = \frac{M}{\nu}, \quad (17)$$

which is equivalent to the results shown in [7], where the constraint of PUSCH resources is ignored. Note that since $\mathbb{E}[S|p,\nu]$ is a concave function due to $d^2\mathbb{E}[S|p,\nu]/dp^2 < 0$, ACB factors should be always set to be equal to or less than $p^* = \frac{M}{\nu}$. However, when $K < M$, it is difficult to find the closed-form expression for the optimal value of p^*. Hence, we can approximate $\mathbb{E}[T|p,\nu]$ as

$$\mathbb{E}[T|p,\nu] \approx \mathbb{E}[S|p,\nu]\cdot\min\left\{1, \frac{K}{\mathbb{E}[D|p,\nu]}\right\}, \quad (18)$$

In case of $\mathbb{E}[D|p^*,\nu] \le K < M$, the ACB factor can be the same as p^*. However, in opposite case of $\mathbb{E}[D|p^*,\nu] > K$, $\mathbb{E}[T|p,\nu]$ increases until $\mathbb{E}[D|p,\nu] = K$, then, it decreases since $\mathbb{E}[S|p,\nu] < \mathbb{E}[D|p,\nu]$ and $\mathbb{E}[D|p,\nu]$ is an increasing function. It implies that p should satisfy $\mathbb{E}[D|p,\nu] = K$. In other words, we need to control all of RA-attempting nodes to obtain PUSCH resource grants in order to maximize the access throughput. As a result, we have the following approximated ACB factor

$$\tilde{p} = \min\left\{1, -\ln\left(1-\frac{K}{M}\right)\right\}\frac{M}{\nu}. \quad (19)$$

Fig. 1 shows an example of average access throughput when $K = 20$ and $\nu = 80$. More specifically, on average, for $p \le \tilde{p}$, the eNodeB can allocate PUSCH resources to all of detected preambles, and then, $\mathbb{E}[S|p,\nu]$ nodes can succeed in RA. On the other hand, for $p > \tilde{p}$, the eNodeB only chooses K preambles among $\mathbb{E}[D|p,\nu]$ preambles and allocates PUSCH resources to them, and then, on average $K\cdot\mathbb{E}[S|p,\nu]/\mathbb{E}[D|p,\nu]$ nodes can succeed in RA on a single PRACH slot. It is worth noting that $\mathbb{E}[T|p^*,\nu] - \mathbb{E}[T|\tilde{p},\nu]$ is negligible. Algorithm 1 summarizes the computation algorithm for the ACB factors. Estimation method of ν will be explained in the next Section.

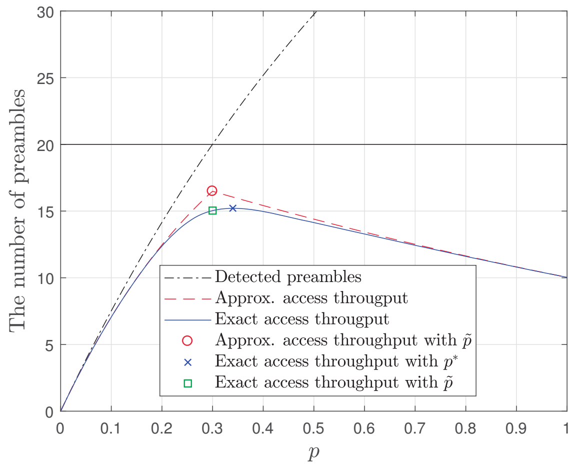

Fig. 1. The number of detected preambles, the exact access throughput, and the approximated access throughput when $K = 20$ and $\nu = 80$.

Algorithm 1 Computation algorithm for the ACB factor

1: Given M preambles and K allocable PUSCH resources.
2: Estimate the number of backlogged nodes ν.
3: **if** $K = M$ **then**
4: $\quad p = p^* = \min\{1, \frac{M}{\nu}\}$
5: **else**
6: $\quad p = \tilde{p} = \min\left\{1, \min\left\{1, -\ln\left(1-\frac{K}{M}\right)\right\}\frac{M}{\nu}\right\}$.
7: **end if**

IV. Bayesian Estimation and Update of ν

Before generating an ACB factor, the number of backlogged nodes ν should be estimated. In order to estimate ν, we utilize the Bayesian estimation and update algorithm based on the observations of the number of undetected (idle) preambles [7]. After nodes attempt RAs by transmitting preamble signals according to the ACB factor p, which is computed based on the estimated value of ν, K, and M, the eNodeB can exactly identify the number of undetected preambles r during the preamble detection procedure. It is notable that at the first step, the eNodeB cannot distinguish the collision or not for the detected preambles. Given this information r, we can correct the estimated value of ν by the estimation offset $\Delta\nu = \mathbb{E}[n|r,p,\nu] - \nu$, where $\mathbb{E}[n|r,p,\nu]$ is written as [7]

$$\mathbb{E}[n|r,p,\nu] = \frac{\sum_{n=0}^{\infty} n\Pr\{n,r|p,\nu\}}{\Pr\{r|p,\nu\}} = \nu\left[(1-p) + \frac{p\left(1-\frac{r}{M}\right)}{\left(1-e^{-\frac{p\nu}{M}}\right)}\right]. \quad (20)$$

Algorithm 2 Estimation and update algorithm of ν

1: Initialize $\nu_0 = M$ and $k_0 = 0$.
2: **if** $p_{i-1} = p^*_{i-1}$ **then**
3: $\quad \Delta = \frac{Me^{-1}-r}{1-e^{-1}}$
4: **else if** $p_{i-1} = \tilde{p}_{i-1}$ **then**
5: $\quad \Delta = -\ln\left(1-\frac{K}{M}\right)\left(\frac{M}{K}\right)(M-K-r)$
6: **end if**
7: $\nu_{i-1} = \nu_{i-1} + \Delta\nu$ ▷ Update of ν
8: **if** $\Delta\nu > 0$ **then**
9: $\quad k_i = k_{i-1} + 1$ and $\nu_{i-1} = \nu_{i-1} + k_i \cdot \Delta\nu$ ▷ Boosting
10: **else**
11: $\quad k_i = 0$ and $\nu_{i-1} = \nu_{i-1}$
12: **end if**
13: $\nu_i = \max(1, \nu_{i-1} - c_{i-1})$ ▷ New estimation of ν for slot i

TABLE I
SIMULATION PARAMETERS AND VALUES

Parameters	Values
The number of preambles, M	64
The number of allocable PUSCH resources, K	$20-64$
PRACH interval time, T_{interval}	1 ms
Activation duration, T_{act}	500 ms
The number of activation slots, I_{act}	500 slots
Total number of RA-attempting nodes, N	10000
Bursty traffic parameters, α and β	3 and 4

In particular, we have estimation offsets:

$$\Delta\nu = \mathbb{E}[n|r,p,\nu] - \nu = \nu p \left(\frac{e^{-\frac{p\nu}{M}} - \frac{r}{M}}{1-e^{-\frac{p\nu}{M}}}\right), \tag{21}$$

$$\Delta\nu|_{p=p^*} = \frac{Me^{-1}-r}{1-e^{-1}}, \tag{22}$$

$$\Delta\nu|_{p=\tilde{p}} = -\ln\left(1-\frac{K}{M}\right)\left(\frac{M}{K}\right)(M-K-r), \tag{23}$$

where $p^* = \frac{M}{\nu}$ and $\tilde{p} = -\ln\left(1-\frac{K}{M}\right)\frac{M}{\nu}$.

Algorithm 2 summarizes the estimation and update algorithm of ν. The eNodeB first estimates ν_0 as the number of preambles M. Following [7], we also introduce a boosting factor k_i in order to consider bursty traffic. It implies that when we observe that the traffic continuously increases based on $\Delta\nu > 0$, we boost the estimation ν. In line 13, the new estimation of ν_i is calculated by $\nu_i = \nu_{i-1} - c_{i-1}$, where c_{i-1} denotes the number of RA success nodes on the $(i-1)$th PRACH slot.

V. NUMERICAL RESULTS

Table I summarizes the simulation parameters and values. We assume that PRACH appears every 1 ms, i.e, $T_{\text{interval}} = 1$ ms. 10000 machine nodes attempt to connect to the network in a bursty manner through the four-step RA procedure as varying PUSCH resources, K, from 20 to 64. We evaluate the performance of the proposed mechanism in terms of the total service time, the average RA delay, and the energy efficiency, compared to the Jin's RA control scheme [7] and the ideal scheme. The ideal scheme follows the same RA control mechanism as the proposed mechanism, but it is assumed to exactly know the number of backlogged nodes on every slot.

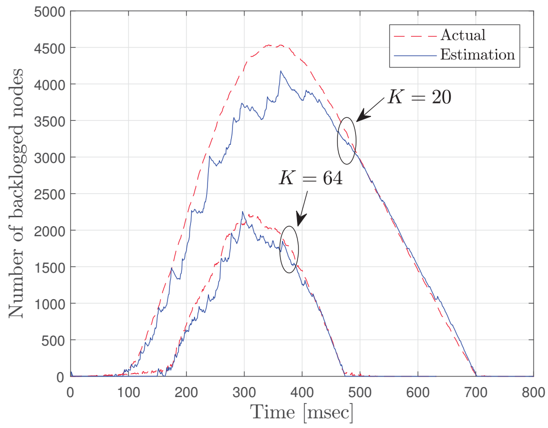

Fig. 2. Estimation for 10000 backlogged machine nodes with $K = 20$ and $K = 64$.

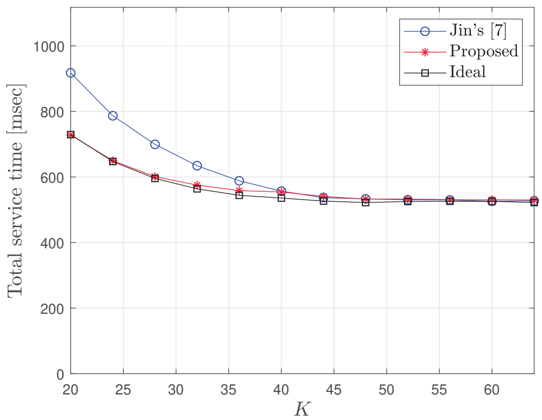

Fig. 3. Total service time.

Fig. 2 shows the Bayesian estimation result for the number of backlogged nodes with sufficient PUSCH resources ($K = 64$) and insufficient PUSCH resources ($K = 20$), respectively. Even though there exist a gap between the estimation and actual values, the estimation algorithm keeps well track of the actual values especially during the traffic descent period. Due to the backlog estimation errors, the proposed RA control mechanism may show the degraded performance, compared to the ideal mechanism.

Fig. 3 shows the total service time for varying the number of allocable PUSCH resources K. The proposed RA control mechanism generates the ACB factors by considering both of the available preambles and the allocable PUSCH resources, while the Jin's scheme only considers the available preambles. Thus, when the PUSCH resources are insufficient compared to the number of detected preambles, e.g., $K < 40$, the proposed mechanism takes shorter time for all nodes to complete the RA service, compared to the Jin's scheme. In terms of the total service time, the proposed mechanism shows the similar result to that of the ideal scheme.

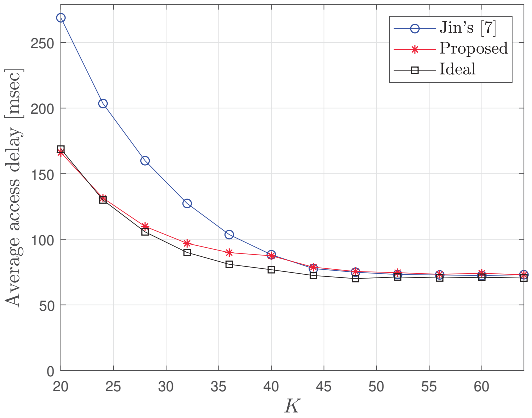

Fig. 4. Average random access delay.

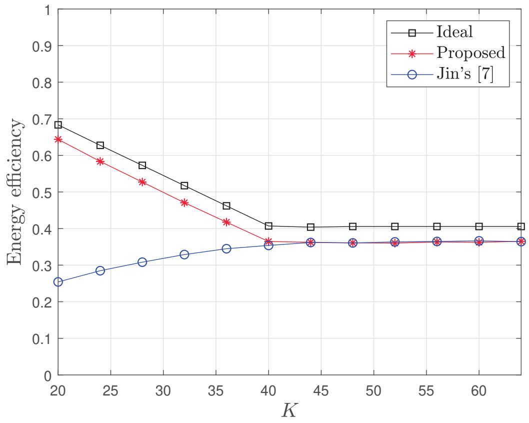

Fig. 5. Energy efficient when $E_{S1} = E_{S3}$.

Fig. 4 shows the average RA delay for varying the number of allocable PUSCH resources K. We can observe a huge difference of average RA delay between the proposed mechanism and the Jin's scheme. In case of $K = 20$, the proposed mechanism shows the average RA delay of 168 ms, while the Jin's scheme shows the average RA delay of 268 ms.

Lastly, Fig. 5 shows the energy efficiency for varying the number of allocable PUSCH resources K, where we assume that the uplink transmission energy for RA-step 1 and 3 is identical, i.e., $E_{S1} = E_{S3}$. The energy efficiency of the proposed mechanism decreases from approximately 0.65 to 0.36, while the energy efficiency of the Jin's scheme increases from approximately 0.25 to 0.36. The result implies that we should consider the allocable PUSCH resources in order to control massive energy-critical devices for successful RAs.

VI. Conclusion

In this paper, we proposed an RA control mechanism for massive IoT networks. Specifically, we utilized the Bayesian backlog estimation algorithm to estimate the number of backlogged nodes, and based on this information, the optimized ACB factor is computed by considering the available preambles and the allocable PUSCH resources in order to maximize the access throughput. Through extensive computer simulations, we evaluated the performance of the proposed mechanism in terms of the total service time, the average RA delay, and the energy efficiency. The proposed mechanism outperforms the conventional scheme when the PUSCH resources are insufficient, which implies that RA control should carefully consider not only the available preambles but also the allocable PUSCH resources. Especially, the proposed mechanism shows a higher energy efficiency than the conventional scheme, where the energy efficiency is very important factor for battery-constrained IoT devices.

Acknowledgment

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2018R1C1B6008126) and in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2017-0-01635) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

[1] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-things-based smart cities: Recent advances and challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, Sept. 2017.

[2] M. Tavana, V. Shah-Mansouri, and V. W. Wong, "Congestion control for bursty M2M traffic in LTE networks," in *Proc. IEEE ICC*, June 2015, pp. 5815–5820.

[3] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, 2013.

[4] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[5] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 4–16, First Quarter 2014.

[6] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9847–9861, Dec. 2016.

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

[8] P. Osti, P. Lassila, S. Aalto, A. Larmo, and T. Tirronen, "Analysis of PDCCH performance for M2M traffic in LTE," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4357–4371, Nov. 2014.

[9] H. S. Jang, H.-S Park, and D. K. Sung, "A non-orthogonal resource allocation scheme in spatial group based random access for cellular M2M communications," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4496–4500, May 2017.

[10] H. S. Jang, B. C. Jung, and D. K. Sung, "Dynamic access control with resource limitation for group paging-based cellular IoT systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5065–5075, Dec. 2018.

[11] *Medium Access Control (MAC) protocol specification (Release 14)*, 3GPP TS 36.321 V14.6.0, Mar. 2018.

[12] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons Ltd., 2009.

[13] 3GPP, "[70bis#11]-LTE: MTC LTE simulations," *3GPP TSG-RAN WG2 meeting# 71 R2-104663*, Aug. 2010.