electronics

MDPI

Article

Combined Access Barring Scheme for IoT Devices Using Bayesian Estimation

Waqas Tariq Toor [1], Maira Alvi [1] and Mamta Agiwal [2,*]

[1] Department of Electrical Engineering, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Punjab 64200, Pakistan; waqas.toor@kfueit.edu.pk (W.T.T.); maira.alvi@kfueit.edu.pk (M.A.)

[2] Department of Electrical Engineering, Sejong University, Seoul 05006, Korea

* Correspondence: mamta@sejong.ac.kr

Received: 17 November 2020; Accepted: 14 December 2020; Published: 18 December 2020

Abstract: This paper focuses on proposing a new access barring scheme for internet of things (IoT) devices in long term evolution advanced (LTE/LTE-A) and 5G networks. Massive number of IoT devices communicating simultaneously is one of the hallmarks of the future communication networks such as 5G and beyond. The problem of congestion also comes with this massive communication for which access barring is one of the solutions. So, it is required that sophisticated access barring techniques are designed such that the congestion is avoided and these devices get served in less time. Legacy access barring schemes like access class barring (ACB) and extended access barring (EAB) suffer from high energy consumption and high access delay respectively. However, our proposed scheme provides less energy consumption than ACB while giving less access delay than EAB. The proposed scheme maximizes the success probability while reducing the number of collisions at the same time. The scheme is based on an approximation of the number of IoT devices based on details available to the eNodeB of the number of idle, successful and collided preambles. Extensive Matlab simulations are performed to validate our claims and analysis.

Keywords: access class barring; extended access barring; IoT; total service time; 5G communications

1. Introduction

Internet of things (IoT) is one of the technologies to shape the future in communications [1]. Huge number of devices are expected to communicate autonomously to each other, or to the server. It is shown in Figure 1 that according to [2], the count of devices that desire to connect to the network will increase to in excess of 75 billion by the end of 2025. IoT finds its role in various applications that are to be used in near future, for example smart grids, smart hospitals and even smart cities. The key idea behind all these applications is that the generated data, either from sensors, or other smart devices, is transmitted to the central stations. Similarly in some applications, the devices may be required to send data to their peer devices for cooperation or decision making. Thus, the nature of traffic is shifting from human type communications (HTC) to machine type communication (MTC) or IoT. HTC is generally characterized with sparse traffic and long traffic duration whereas in IoT, communicating devices are in large number and the nature of traffic is bursty. The devices generally come in bursts and the duration of traffic is short.

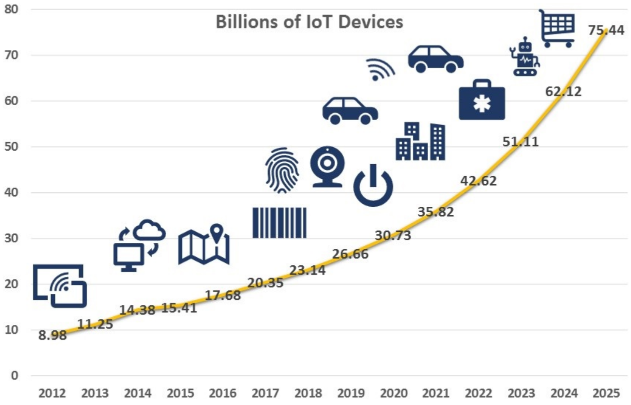

Figure 1. Internet of things (IoT) devices in billions over the years.

Long term evolution (LTE/LTE-A) promises to overcome the problems of the bursty nature of IoT traffic. Recognizing different requirements of IoT, LTE has maintained a separate procedure to deal with this traffic. First step in dealing with IoT devices is the random access (RA) process. The devices have to undergo RA process to get scheduled service from the network. Since the resources to deal with RA process are limited and the number of devices attempting RA process is large, serious congestion can take place. So, the devices will be unable to complete RA process successfully and hence cannot get served through scheduled access by the network. In literature, many solutions are proposed for this problem [3–7]. In order to address the issue of congestion, access barring (AB) is considered as a accepted technique by third generation partnership project (3GPP). Access barring is the process in which some devices are barred from transmission based on some pre-defined procedure. In LTE-A, the most sought solutions for congestion control are access class barring (ACB) [8] and extended access barring (EAB) [9]. While EAB is designed only for MTC/IoT traffic, ACB can handle both IoT and HTC traffic.

In ACB, the reduction in congestion is proportional to the reduction in number of collisions, or in other words, to the probability of success. The success probability in turn depends upon the transmission probability of the devices. The process of ACB, in short, can be explained as follows: In ACB, a transmission probability called ACB factor is broadcast by eNodeB. Then, every device generates a random number and then transmits if the random number is less than the probability broadcast by the eNodeB, otherwise the device refrains from transmission. The devices that become eligible for transmission are said to have passed ACB check. This ACB factor is of prime importance in controlling congestion in IoT traffic.If we take the value of ACB factor low, very less number of IoT devices pass the ACB check and resources may go idle. On the other hand if the ACB factor is high, large number of devices may pass the ACB check, increasing the number of collisions. So the ACB factor needs to be optimal.

In [8], the authors propose the methodology to obtain the optimal ACB factor that rely on the number of active devices in the network. In [10], the authors used fixed ACB factor, i.e., they assume fixed number of devices in the network. However, we can observe that the number of IoT devices is always changing. In [11], the authors use the information on number of idle, success and collided preambles (channels), and maximum likelihood to find the number of devices in the network. Then the authors use Kalman filter to fine tune their estimation. But the maximum likelihood is an expansive operation and not good to use in real time scenarios. In [12], the authors also use the information on collided preambles for the last three slots, and use heuristics in devising algorithm to find the number of devices, and in turn the optimal transmission probability. But the algorithm fails if the heuristics show slight deviation. In [13] the authors use information on success and idle preambles and employ pseudo Bayesian algorithm to find the number of devices in the network. The authors develop a simple update rule to estimate the number of devices, which can be implemented easily in real time scenario.

In EAB, each device is assigned a class number and eNodeB only allows transmission from devices of only one class. The devices from other classes are forced to wait until their class is unbarred. In [9], the authors take into account the specifications suggested by 3GPP and presented analytical model for EAB and its performance. In [14], the authors study the parameters of EAB and compares its performance with ACB. The authors find out that ACB has higher collision probability than EAB while EAB has higher access delay than ACB. More number of collisions means more retransmissions and in turn more energy usage. Thus the devices that are delay constrained cannot use EAB while ACB is used by the devices that are not energy contrained. So we need to devise a strategy which can be adopted by both delay and energy constrained IoT devices.

In [15], we presented a combined access barring (CAB), a combination of ACB and EAB, in which ACB is applied within EAB. CAB presents less collision probability than ACB and less access delay than EAB. In this paper we extend our analysis and improve the performance of the proposed technique in [15] by introducing practicality. In [15], it is assumed that the number of devices in the network is known which is not true in real time scenarios. Moreover, the number of devices used in the algorithm are the devices present in the whole network. However, the number of devices that need to be estimated is the devices within a class that is allowed to transmit. The rest of the paper is organized as follows. In Section 2, we present the random access process in LTE-A and associated ACB and EAB process. The proposed technique CAB, is presented in Section 3. Performance evaluation and discussion are presented in Section 4 and finally the paper is concluded in Section 5.

2. Random Access Procedure in LTE-A

As mentioned earlier that the first step in getting access to the LTE-A network is the random access (RA). RA in LTE-A has two types namely contention free and contention based. In contention free RA, the devices have priorities and those devices do not contend for the channel. The channel is already reserved for them. But these devices form very small percentage of total number of devices. In practice, the general IoT devices do not have priorities. They need to contend for the channel for RA process to have a success. So that is why we chose contention based RA in our paper. This is where almost all of the devices lie and hence our choice is realistic. We consider only contention based RA in this paper and from here on, RA represent the contention based RA unless specified. RA process in LTE-A/5G is a four step procedure and depends on the exchange of four messages between IoT device and eNodeB as mentioned in release 15 [16]. As mentioned in release 16 [17], in 5G, the RA procedure consists of two steps in which 1st and 3rd message of four step process are combined to make 1st message of two step RA process. Then 2nd and 4th message of four step RA are combined to make 2nd message of two step RA process. Two step RA procedure helps in reducing delay in ultra reliable low latency communication (URLLC) applications [18]. As our proposed scheme can be applied any of the two, here we consider four step RA procedure. The eNodeB broadcasts $(R \leq 64)$ RA preambles and then the four steps among IoT devices and eNodeB are as follows:

- **RA preamble transmission:** At this step, each IoT device selects one of the R RA preambles and transmits it to the eNodeB along with its RA radio network temporary identity (RA-RNTI). eNodeB use this identity to further communicate with the IoT devices. It is possible that two or more IoT devices may select the same preamble and at this step, eNodeB is unaware of this scenario.
- **RA response (RAR) message reception:** At this step eNodeB sends RAR message to the IoT device in response to the reception of Msg 1 from the corresponding device. If RAR is not received by the IoT device within a certain time (12 ms at maximum), it sends Msg 1 again. In addition, eNodeB assigns temporary cell RNTI (C-RNTI) to the IoT device for furhter communication. Other information such as timing advance, uplink grant resource etc is also provided to IoT device from eNodeB at this step.
- **Connection request message transmission:** If RAR is received and decoded successfully by IoT device, it sends connection request message to eNodeB. This message contains temporary mobile

subscriber identity (TMSI) along with other information. TMSI is requied by eNodeB for the transmission of the next message because it is possible that C-RNTI is assigned to more than one IoT devices by eNodeB if these devices have selected same preamble in first step.

- **Contention resolution message reception:** If connection request message is received and decoded successfully by the eNodeB, it sends contention resolution to the IoT device. If contention resolution message is received and decoded by the IoT device, then RA process is said to have completed successfully after which scheduled access can take place. The whole process is presented in Figure 2.

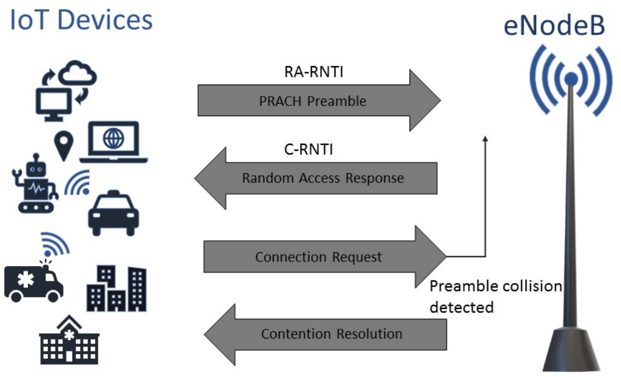

Figure 2. Random Access Procedure.

It is possible that more than one, say two, devices select same preamble. Then when both the IoT devices send connection request message to eNodeB, the eNodeB is unable to decode the message successfully and a collision happens. It is assumed that in the case of collision, eNodeB is not able to decode connection request message from either of the two devices. The eNodeB does not send contention resolution message to any of the collided IoT devices. After time out of their respective timers, the collided IoT devices declare that collision has happened and then start from Msg 1 again. Please note that eNodeB is unable to detect collision till reception of connection request message. However, the authors in [19] show that by using their technique eNodeB is able to detect collision after the reception of RA preamble message (Msg 1). This technique is called early preamble collision detection.

2.1. Access Class Barring

As described above that IoT devices choose one of $R \leq 64$ preambles to transmit and the number of IoT devices may be large. The probability of the event that two or more devices select the same preambles is quite large and thus as a result collisions may happen. This results in loss of time and frequency resources. Now, ACB process works as follows. Along with R preambles, the eNodeB also broadcasts the ACB factor. Before transmission of RA preamble to eNodeB, each IoT device generates a random number. The device transmits if the generated random number is less than ACB factor and refrains from transmission otherwise. Thus by adopting ACB, the devices that send RA preamble to eNodeB becomes less and manageable. So the ACB factor affects the Msg 1 only and the rest of RA messages remain the same. As mentioned earlier that the ACB factor needs to be optimal in order to overcome the under usage of resources and large number of collisions. The effect of choosing ACB factor is demonstrated in Figure 3. We can see that by choosing ACB factor to be small the number of idle preambles becomes large, whereas the number of collided preambles becomes large if ACB factor is chosen to be too large. In both the cases the number of success preambles is small. However when optimal ACB factor is selected, the number of success preambles become larger than idle and collided preambles.

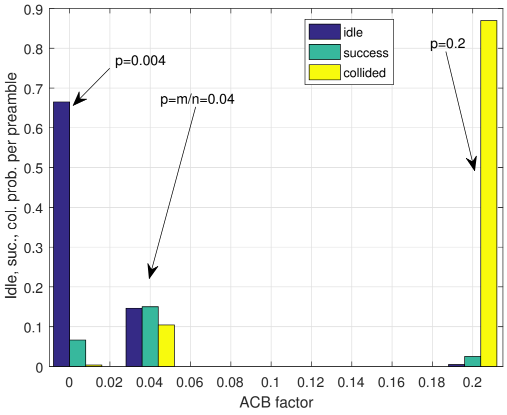

Figure 3. Importance of choosing optimal access class barring (ACB) factor: No. of devices, $U = 1000$, No. of preambles, $R = 40$, simulation time $= 10^6$ slots for fixed probabilities, $p = 0.004, 0.2$, and 0.04 (optimal).

2.2. Extended Access Barring

In EAB, each IoT device is assigned a class number ranging from 0 to 9 and the devices affiliated with only one class can transmit while the other classes are barred. The eNodeB broadcasts a 10-bit bitmap with all bits equal to 0 except one. The index of the bit value 1 indicates the the class number which is unbarred. For example bitmap 0010000000 represents that class number 2 is unbarred while all the other classes are barred. So by allowing only one class among 10 to transmit, the number of collisions can be reduced. Even if the devices of only one class is allowed to transmit, their may be large number of devices within a single class and that also can lead to collisions and then congestion. For coping this issue, EAB also comes with discontinuous transmission (DTX) in which all the devices within one unbarred class do not transmit simultaneously. Each IoT device has its own paging frame (PF) and within paging frame its own paging occasion (PO). Each IoT device wakes up at its own PO within its own PF and then checks from bitmap whether its class is barred or not. It transmits if its class is unbarred otherwise it sleeps again. So there is quite less chance that two devices within one class have same PF and PO, and hence collision can be controlled. However, the devices have to wait longer to transmit the message. Please note that additional access delay is experienced by the IoT devices due to DTX mode in EAB. Thus in EAB, collisions are reduced at the cost of increased access delay while in ACB, the access delay is reduced at the cost of collisions. Thus we need to have a barring mechanism that reduces delay as well as collisions. So we propose combined access barring (CAB) which is given in the next section.

3. The Proposed Technique: Combined Access Barring

3.1. System Model and Proposed Algorithm

We consider RA process of LTE-A explained in Section 2. The eNodeB broadcasts transmission probability $p \in (0, 1)$ and 10-bit bitmap. The transmission probability is dynamic, i.e., it changes in every time slot depending on the number of active devices in the network denoted by u, where $u \subset U$. U denotes total number of IoT devices within the network. By active devices we mean the backlogged devices within unbarred class. The bitmap can be broadcast through system information block (SIB)-14. Each IoT device is assigned a class number between 0 and 9 which is stored in subscriber identity module (SIM). Each IoT device reads bitmap from SIB-14 and checks whether its class is unbarred or not. If the device is in unbarred class, it generates a random number and compares it with p. If the generated number is less than p, it transmits RA preamble to eNodeB otherwise it waits for the next opportunity. After the transmission attempt, eNodeB observes the number of idle, successful and collided preambles and depending on the outcome it estimates the change in backlog. We denote the number of idle and success preambles as i and s respectively. Based on current backlogged IoT devices, the transmission probability for the next slot is calculated.

Please note that the p_t depends upon the number of IoT devices within the certain class unlike in [13], where the estimation on all the devices is considered. Moreover in [13], the information on only idle preambles used for estimation of devices. However in [19], the authors explained that by early collision estimation, the information on idle as well as success preambles can be used. Hence, in this paper we use the information on idle as well as success preambles. We explain the transmission probability p_t and estimation of devices x_t in the next section.

3.2. Analysis

If the class of an IoT device is unbarred then it generates a random number and transmits if it is less than p_t. From here on, we will omit t from subscript of p_t for notational simplicity. Let us denote the number of backlogged IoT devices within the system by u and the number of devices, that have generated random number less than the transmission probability, by v, then we can write,

$$P\left(v|u\right) = \binom{u}{v} p^{v} \left(1-p\right)^{u-v} \tag{1}$$

where the probability $P(v|u)$ denote the probability that v IoT devices among u have qualified for transmission. A success happens on a certain preamble if it is chosen by only one of the v IoT devices. We can write the probability of success as follows:

$$P\left(S_m\right) = \sum_{u=0}^{\infty} \sum_{v=0}^{u} P\left(S_m|v\right) P\left(v|u\right) \tag{2}$$

where $P(S_m|v)$ denotes the probability that there is a success on preamble m, when v devices are allowed to transmit, and can be written as:

$$P\left(S_m|v\right) = \binom{v}{1} \frac{1}{R} \left(1-\frac{1}{R}\right)^{v-1} \tag{3}$$

where R denotes the number of random access preambles. Now putting (1) and (3) in (2) and then solving, we get:

$$P\left(S_m\right) = \frac{up}{R} \left(1-\frac{p}{R}\right)^{u-1} \tag{4}$$

It is important to mention that on a single random access preamble, only one IoT device can get success, so the expected number of devices that get success on preamble m can be written as:

$$E[S_m] = 1 \cdot P(S_m) + 0 \cdot (1-P(S_m)) = \frac{up}{R} \left(1-\frac{p}{R}\right)^{u-1} \tag{5}$$

Then the average number of IoT devices over all R RA preambles can be written as:

$$E[S] = up\left(1-\frac{p}{R}\right)^{u-1} \tag{6}$$

To calculate the optimal transmission probability p that maximizes the success probability and $E[S]$, we need to solve the following optimization problem:

$$\max_{p_{opt}} \left[up\left(1-\frac{p}{R}\right)^{u-1} \right] \tag{7}$$

where p is the optimal transmission probability that maximizes (7) which can be solved by taking derivative and equating it to zero. Then we can have the following relation for optimal transmission probability:

$$p = \min\left(1, \frac{R}{u}\right) \tag{8}$$

The whole procedure is shown in Algorithm 1 where it is assumed that the number of backlogged devices is known.

Algorithm 1 Combined Access Barring (CAB) Scheme with Backlog Known

```
Extract bitmap from SIB-14, $p = 1$
if the class is unbarred then
    Generate random number, $G \in (0, 1)$
    if $G < p$ then
        Choose preamble among $R$ and transmit
    end if
end if
$p = \min\left(1, \frac{R}{u}\right)$        % To be used for the next time slot
Repeat the process from Step 2 in every time slot
```

3.3. Estimation on u

As already explained that the eNodeB does not have the information on the number of active devices u, in order to calculate optimal p and in turn maximize success probability, the eNodeB needs to have estimation on u. The eNodeB can observe the number of successful and idle preambles in each time slot, then the eNodeB needs to calculate $E[u|(i,s)]$ which can be written as:

$$E[u|(i,s)] = \sum_{u=0}^{\infty} uP(u|(i,s)) \tag{9}$$

where $P(u|(i,s))$ is the probability of having u active users if the number of idle and successful preambles are i and s respectively. This probability can be written as follows:

$$P(u|(i,s)) = \frac{P(u,i,s)}{P(i,s)} \tag{10}$$

where $P(i,s)$ is the probability that i idle and s successful preambles are transmitted and can be written as:

$$P(i,s) = \sum_{u=0}^{\infty} P(u,i,s) \tag{11}$$

We can write $P(u,i,s)$ as follows:

$$P(u,i,s) = P((i,s)|u)P(u) = \sum_{v=0}^{u} P((i,s)|v)P(v|u)P(u) \tag{12}$$

$P((i,s)|u)$ in (12) is the aposteriori probability in Bayesian rule while $P(u)$ is the apriori probability. After taking $P(u)$ as apriori probability, the information is then used to get the observation on i and s. To evaluate (12) in each time slot, $P(u)$ is required for each time slot. With increase in the number of IoT devices, finding $P(u)$ for each user is quite time consuming and computationally inefficient. So we assume $P(u)$ to be Poisson distributed and can be written as:

$$P(u) = \frac{x^u}{x!}e^{-x} \tag{13}$$

where $x = E[P(u)]$ is the mean of the Poisson distribution. Now we can rewrite (2) as follows:

$$P\left(S_m\right) = \sum_{u=0}^{\infty} \sum_{v=0}^{u} P\left(S_m|v\right) P\left(v|u\right) P(u) = \frac{xp}{R} e^{-\frac{xp}{R}} \tag{14}$$

Similarly (6) and (8) can be written as follows:

$$E[S] = xpe^{-\frac{xp}{R}} \tag{15}$$

$$p = \min\left(1, \frac{R}{x}\right) \tag{16}$$

We can see from (16) that the estimation on $P(u)$ comes down to estimating the mean of the backlog distribution or x which means that in each time slot, instead of finding exact probability distribution, we find the mean of the distribution. Finding the mean of distribution instead of actual distribution $P(u)$ is termed as pseudo-Bayesian instead of Bayesian [20]. Now we need to find $P(u, i, s)$ and $P(i, s)$ in order to evaluate $P(u|(i, s))$ and $E[u|(i, s)]$ as in (10) and (9). Let us now find $P((i, s)|v)$ to evaluate (12) and (11). This probability depends on choosing i idle and s success preambles among R. Among R there are $\binom{R}{i+s}$ ways to choose idle and success preambles and among $i + s$ there are $\binom{i+s}{s}$ ways to select success preambles among $i + s$. Since for success, there must be one IoT devices for one preamble, then there are $\binom{v}{s}$ ways to choose successful IoT devices among v and then these successful devices can be randomly chosen in $s!$ ways. If s preambles are successful among R, then the remaining $R - i - s$ should neither have success nor they can remain idle. Combining all these statements, we can write the expression of $P((i, s)|v)$ as follows:

$$P((i,s)|v) = \frac{1}{R^v}\left[\binom{R}{i+s}\binom{i+s}{s}\binom{v}{s} s!\,(R-i-s)^{v-s}\right] P_{col}(R-i-s, v-s) \tag{17}$$

where R^v is the number of every possible ways for v IoT devices to select among R preambles randomly and $P_{col}(R - i - s, v - s)$ is the probability of collision in $R - i - s$ channels when $v - s$ IoT devices try to transmit. Let us denote the event F_a be the event and $P(F_a)$ its corresponding probability that a-th of the $R - i - s$ channels is selected by at the most one IoT device, then we can write $P_{col}(R - i - s, v - s)$ as:

$$P_{col}(R-i-s, v-s) = 1 - P\left(\bigcup_{a=1}^{R-i-s} F_a\right) \tag{18}$$

where $P\left(\bigcup_{a=1}^{R-i-s} F_a\right)$, by using inclusion-exclusion principle [21], can be written as follows:

$$P\left(\bigcup_{a=1}^{R-i-s} F_a\right) = \sum_{b=1}^{R-i-s} (-1)^{b+1} S_b(R-i-s, v-s) \tag{19}$$

where $S_b(R - i - s, v - s)$ represents the probability that b out of $R - i - s$ preambles are selected by at the most $v - s$ devices. We can evaluate (19) by using portion of (17) as follows:

$$S_b(R-i-s, v-s) = \sum_{c=0}^{b} \frac{b!}{(R-i-s)^b}\binom{R-i-s}{b}\binom{b}{c}\binom{v-s}{c}\left(1 - \frac{b}{R-i-s}\right)^{v-s-c} \tag{20}$$

Substituting (20) into (19) gives:

$$P_{col}(R-i-s, v-s) = \sum_{b=1}^{R-i-s} \sum_{c=0}^{b} (-1)^b \frac{b!}{(R-i-s)^b}\binom{R-i-s}{b}\binom{b}{c}\binom{v-s}{c}\left(1 - \frac{b}{R-i-s}\right)^{v-s-c} \tag{21}$$

Now we can plug in (21) in (17) to calculate $P((i,s)|v)$. After evaluating $P((i,s)|v)$, we can evaluate (11) as follows [22]

$$\begin{aligned}P(i,s) &= \sum_{u=0}^{\infty}\sum_{v=0}^{u}\sum_{b=0}^{R-i-s}\sum_{c=0}^{b}\frac{(-1)^b s! b!}{R^v (R-i-s)^b}\left[\binom{R}{i+s}\binom{i+s}{s}\binom{v}{s}(R-i-s)^{v-s}\right] \\ &\times \binom{R-i-s}{b}\binom{b}{c}\binom{v-s}{c}\left(1-\frac{b}{R-i-s}\right)^{v-s-c}\binom{u}{v}p^v(1-p)^{u-v}\frac{x^u}{x!}e^{-x} \\ &= \binom{R}{i+s}\binom{i+s}{s}\left(\frac{px}{R}\right)^s e^{-px}\left(e^{-\frac{px}{R}}-1-\frac{px}{R}\right)^{R-i-s} \end{aligned} \tag{22}$$

Using (22) along with (10), we can evaluate (9) as follows:

$$E[u|(i,s)] = px + x + s + \frac{px(R-i-s)\left(e^{\frac{px}{R}}-1\right)}{\left(Re^{\frac{px}{R}}-R-px\right)} \tag{23}$$

The change in backlog can be calculated as follows:

$$\Delta x = E[u|(i,s)] - x \tag{24}$$

Now, if we put optimal p from (16) in (24) through (23), and then dividing it by 10, we can get Δx as follows:

$$\Delta x = 0.13922 \times R - 0.23922 \times i - 0.13922 \times s \tag{25}$$

The division by 10 is because of the assumption that the incoming devices are uniformly distributed in all of the 10 classes. The procedure involving backlog estimation is shown in Algorithm 2 with γ is used as boosting factor to incorporate bursty nature of traffic [13]. In every slot, x is updated to find optimal p by updating Δx in each time slot.

Algorithm 2 Combined Access Barring (CAB) Scheme using Bayesian Estimation

```
1:  Extract bitmap from SIB-14, $p = 1, x = 0$
2:  if the class is unbarred then
3:      Generate random number, $G \in (0,1)$
4:      if $G < p$ then
5:          Choose preamble among $R$ and transmit
6:      end if
7:  end if
8:  Observe the number of idle and success preambles among $R$
9:  if $i$ idle and $s$ successful preambles then
10:     $\Delta x = 0.13922 \times R - 0.3922 \times i - 0.13922 \times s$
11:     if $\Delta x > 0$ then
12:         $\gamma = \gamma + 1$
13:     else
14:         $\gamma = R$
15:     end if
16:     $x_{t-1} = x_{t-1} + \gamma \times \Delta x$
17:     $x_t = x_t + \Delta x$
18: end if
19: $x_t$
20: $x_t = \max(R, x_t)$
21: $p = \min\left(1, \frac{R}{x_t}\right)$        % To be used for the next time slot
22: Repeat the process from Step 2 in every time slot
```

4. Performance Evaluation and Discussion

The performance evaluation of the proposed technique is verified by performing extensive simulations in Matlab. System model is programmed in Matlab and the results are plotted that are shown and explained in this section. The general parameter setting for the simulations is mentioned in Table 1. The number of devices is U = 10,000, 30,000 and it affects the overall TST because more number of devices means larger TST. Since all the devices arrive in the system within very short time $T_A = 500$ ms, the access delay also depends on the total number of devices. The simulation runs until all the U devices get served. The number of preambles are varied from $R = 5 \rightarrow 50$ and the selection also affects TST and access delay. Larger the number of preambles, larger is the choice for the devices to choose a preamble from a given pool. Then the probability of collision decreases, and consequently the probability of success (or idle) increases and hence TST and access delay is reduced. The periodicity of bitmap T_P is also an important factor as the eNodeB changes the class access based on this parameter. Larger the value of T_P, larger is the waiting time of devices belonging to this class and vice versa, and its value is taken to be 320 ms as it can have value of $160, 320, 640, 1280$ ms [23]. To analyze the performance of the proposed scheme, we take Beta distribution as arrival distribution which can be written as follows:

$$f_T(t) = \frac{t^{\alpha-1}(T_A - t)^{\beta-1}}{T_A^{\alpha+\beta-1} B(\alpha,\beta)}, \tag{26}$$

where $B(\alpha, \beta)$ denotes the Beta function $\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$, $\alpha = 3$, and $\beta = 4$ [16]. T_A is the time in which all the devices arrive in the system, and its value is assumed to be 500 ms. The number of devices arriving in one time slot can be written as:

$$\lambda_i = U \int_{t_{i-1}}^{t_i} f_T(t)dt, \qquad i = 1, 2, \cdots, T_A \tag{27}$$

This distribution is considered to imitate the bursty nature of the arrival pattern which is normally the situation for IoT devices.

Table 1. Performance Parameters.

Parameter	**Value**
Total Number of Devices (U)	10,000, 30,000
No. of Preambles (R)	5 to 50
Periodicity of bitmap broadcast (T_P)	320 ms
Performance Metrics	Access Delay, Total Service Time

In Figure 4, we plot TST versus number of preambles for U = 10,000 using ACB, EAB and proposed CAB schemes. For the proposed CAB scheme, we have plotted the scenarios depicted in Algorithms 1 and 2. Generally we can foresee that TST increases with increase in U and decreases with increase in R which is obvious because large R means large pool of choices for the devices. We see that ACB exhibits the least TST while EAB shows the largest value of TST. The proposed CAB shows less TST than EAB because it does not uses DTX whereas it shows more TST than ACB because the devices need to wait for their classes to be declared unbarred. Moreover, we see that the CAB plot shown by Algorithms 1 and 2 are quite close to each other. This shows that our estimation for the backlog closely follows the actual number of backlogged devices. We also included the simulations of the techniques previously presented in [11–13].

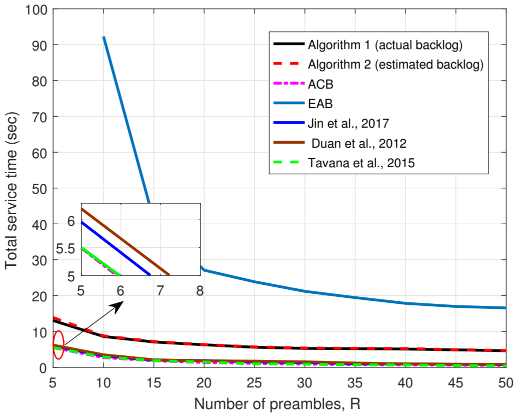

Figure 4. Total service time (TST) vs. number of preambles, *R*.

In Figure 5, we plot the transmission probability for CAB in case of Algorithms 1 and 2. For ACB, at the start, the number of devices are less and the corresponding transmission probability is close to 1. As the number of devices starts increasing, the transmission probability starts decreasing. After the devices get success, the number of devices decreases again thus increasing the transmission probability. But for CAB, the transmission probability decreases after sometime and then starts increasing before coming down again. This is because after periodicity of bitmap transmission, the class which is unbarred already has devices to be served.

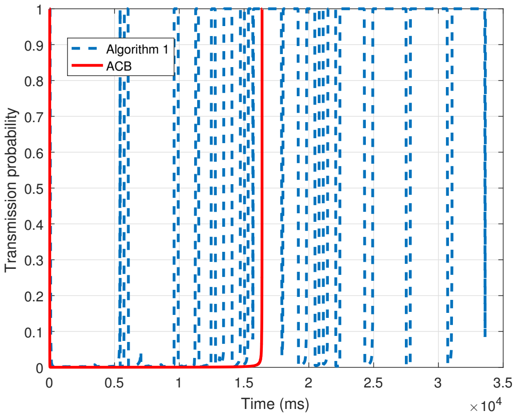

Figure 5. Transmission probability vs. time.

In Figure 6, we plot access delay for U = 30,000 for ACB, EAB and proposed CAB. Again, for the proposed CAB, both Algorithms 1 and 2 based scenarios are plotted.We see that the access delay decreases as the number of preambles increases. We also observe that the access delay shown by CAB is significantly less than EAB but slightly larger than ACB. Moreover the delay calculated by estimation of backlog devices (Algorithm 2) is very close the delay found by observing actual number of devices (Algorithm 1).

As described above that the TST or delay shown by CAB is large as compared to ACB which is due to the periodicity of bitmap. If the bitmap allows all the classes all the time, then CAB becomes ACB. In Figure 7, we plot TST by varying periodicity of bitmap broadcast (T_P). We see that the TST becomes larger and larger as T_P increases as we vary T_P from 160 to 1280 for both Algorithms 1 and 2.

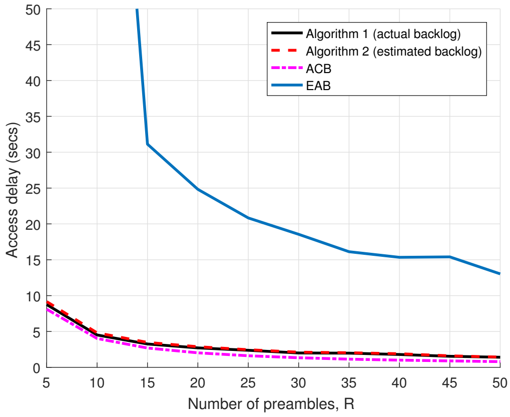

Figure 6. Average access delay vs. number of preambles, *R*.

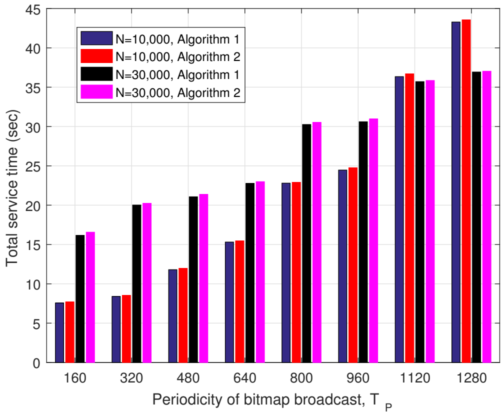

Figure 7. TST vs. periodicity of bitmap broadcast, T_P.

In Figure 8, we plot the number of collisions for ACB, CAB and EAB. We see that the ACB shows largest number of collisions while CAB exhibits much less collisions than ACB. CAB shows slightly larger collisions than EAB which is due to the reason that CAB does not use DTX. Hence we see that for energy sensitive devices, CAB performs much better than ACB because collisions and retransmissions have direct relation with energy usage. Large collisions means large retransmissions and loss of energy. Thus the proposed CAB is useful for energy constrained devices as well.

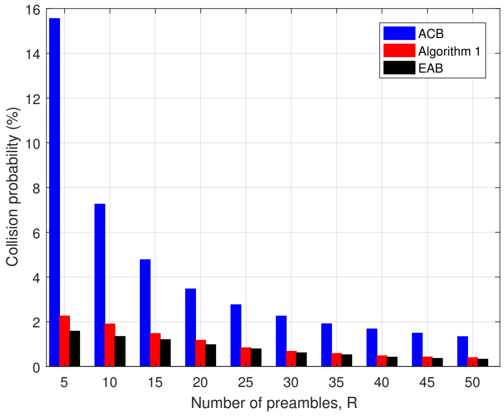

Figure 8. Number of collisions vs. number of preambles, *R*.

Although Figure 8 implicitly shows that the energy efficiency of ACB is the worst as compared to EAB and CAB, we have added Figure 9 for explicitly showing the results of energy efficiency (EE). Instead of expressing EE as energy expended per unit transmission, we find it more useful to present the results on energy wasted per unit transmission (energy inefficiency (EE′) i.e., how much energy is wasted in the collision when it could have been usefully utilized in success. We see that ACB has higher value of energy waste as compared to CAB and EAB, whereas CAB has slightly higher waste of energy than EAB.

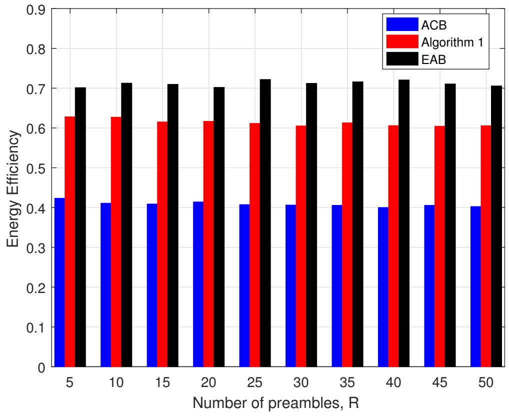

Figure 9. EE′ vs. number of preambles, *R*.

5. Conclusions and Future Works

In this paper we discuss combined access barring (CAB) scheme to deal with the inherent issues of access class barring (ACB) and extended access barring (EAB). ACB suffers from extensive collisions while EAB has problem of large access delays and total service time (TST). ACB is not suitable for energy sensitive devices whereas EAB is not a good choice for delay sensitive devices. CAB provides less collisions in contrast to ACB and less TST and access delay as compared to EAB. Bayesian estimation is used to estimate the number of active devices by observing idle and success preambles, and optimal transmission probability is calculated. Our proposed scheme shows better performance than ACB and EAB in terms of collisions and delay respectively. For the future work, we intend to work on the energy efficiency of our proposed CAB by introducing machine learning (ML) algorithm. By preparing the data set that involves the results on the observations of the preambles being idle, success, or collided, and other parameters such as number of successive trials, we can optimize the transmission time of the IoT devices. By optimizing the transmission time, the devices can have least collisions, and hence the energy used in the re-transmissions can be saved making the system energy efficient.

Author Contributions: W.T.T.: Original idea, Formal analysis, Investigation, Methodology, Writing—original draft, Writing—review & editing. M.A. (Maira Alvi): Formal analysis, Investigation, Methodology, Writing—review & editing. M.A. (Mamta Agiwal): Investigation, Methodology, Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the faculty research fund of Sejong University in 2020.

Conflicts of Interest: There are no conflicts of interest.

References

1. Faizan, Q.; Siddiqui, M.U.A.; Hindia, M.H.D.; Hassan, R.; Nguyen, Q.N. Issues, Challenges, and Research Trends in Spectrum Management: A Comprehensive Overview and New Vision for Designing 6G Networks. *Electronics* **2020**, *9*, 1416.

2. Internet of Things—Number of Connected Devices Worldwide 2015–2025. Available online: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed on 30 November 2020).
3. Condoluci, M.; Dohler, M.; Araniti, G.; Molinaro, A.; Sachs, J. Enhanced Radio Access and Data Transmission Procedures Facilitating Industry Compliant Machine Type Communications over LTE-based 5G Networks. *IEEE Wirel. Commun.* **2016**, *23*, 56–63. [CrossRef]
4. Ali, M.S.; Hossain, E.; Kim, D.I. LTE/LTE-A Random Access for Massive Machine-Type Communications in Smart Cities. *IEEE Commun. Mag.* **2017**, *55*, 76–83. [CrossRef]
5. Chen, Y.-J.; Cheng, L.-Y.; Wang, L.-C. Prioritized Resource Reservation for Reducing RA Delay in 5G URLLC. In Proceedings of the IEEE Personal Indoor and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5.
6. Nwogu, O.A.; Diaz, G.; Abdennebi, M. A Combined Static/Dynamic Partitioned Resource Usage Approach for Random Access in 5G Cellular Networks. In Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19– 21 September 2019; pp. 1–6.
7. 3GPP TR 37.868. RAN Improvements for Machine Type Communications (Release 11). 2011. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2630 (accessed on 4 August 2020)
8. Wang, Z.; Wong, V.W.S. Optimal access class barring for stationary machine type communication devices with timing advance information. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 5374–5387. [CrossRef]
9. Cheng, R.G.; Chen, J.; Chen, D.W.; Wei, C.H. Modeling and analysis of an extended access barring algorithm for machine-type communications in LTE-A networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 2956–2968. [CrossRef]
10. Lee, J.; Guo, J.; Durrani, S. Analytical framework for access class barring in machine type communication. In Proceedings of the IEEE Personal Indoor and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6.
11. Tavana, M.; Shah-Mansouri, V.; Wong, V.W.S. Congestion control for bursty M2M traffic in LTE networks. In Proceedings of the IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5815–5820.
12. Duan, S.; Shah-Mansouri, V.; Wong, V.W.S. Dynamic access class barring for M2M communications in LTE networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 4747–4752.
13. Jin, H.; Toor, W.T.; Jung, B.C.; Seo, J. Recursive Pseudo-Bayesian Access Class Barring for M2M Communications in LTE Systems. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8595–8599. [CrossRef]
14. Toor, W.T.; Jin, H. Comparative study of access class barring and extended access barring for machine type communications. In Proceedings of the International Conference on Technology Convergence (ICTC), Jeju Island, Korea, 18–20 October 2017; pp. 130–132.
15. Toor, W.T.; Jin, H. Combined Access Barring for Energy and Delay Constrained Machine Type Communications. In Proceedings of the International Conference on Technology Convergence (ICTC), Jeju Island, Korea, 17–19 October 2018; pp. 604–609.
16. 3GPP TS 38.321 V15.8.0. Medium Access Control (MAC) Protocol Specification (Release 15). 2019. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3194 (accessed on 2 September 2020).
17. The 5 G Evolution: 3GPP Relases 16–17. Available online: https://www.5gamericas.org/wp-content/uploads/2020/01/5G-Evolution-3GPP-R16-R17-FINAL.pdf (accessed on 29 November 2020).
18. Toor, W.T.; Basit, A.; Maroof, N.; Khan, S.A.; Saadi, M. Evolution of random access process: From Legacy networks to 5G and beyond. *Trans. Emerg. Telecommun. Technol.* **2019**, e3776. [CrossRef]
19. Jang, H.S.; Kim, S.M.; Park, H.S.; Sung, D.K. An early preamble collision detection scheme based on tagged preambles for cellular M2M random access. *IEEE Trans. Veh. Technol.* **2017**, *66*, 5974–5984. [CrossRef]
20. Rivest, R.L. Network control by Bayesian broadcast. *IEEE Trans. Inf. Theory* **1987**, *IT-33*, 323–328. [CrossRef]
21. Ross, S.M. *Stochastic Processes*, 2nd ed.; Wiley: Hoboken, NJ, USA, 1995.

22. Toor, W.T.; Seo, J.B.; Jin, H. Practical splitting algorithm for multi-channel slotted random access systems. *IEEE Trans. Mob. Comput.* **2020**, *19*, 2863–2873. [CrossRef]
23. 3GPP R2-120270. *Further Performance Evaluation of EAB Information Update Mechanisms*; RAN2 77; Intel Corporation: Santa Clara, CA, USA, 2012.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.