Congestion Control for Bursty M2M Traffic in LTE Networks

Morteza Tavana*, Vahid Shah-Mansouri*, and Vincent W.S. Wong†

* School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
† Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada
Email: *{morteza_tavana, vmansouri}@ut.ac.ir, †vincentw@ece.ubc.ca

***Abstract*—In machine to machine (M2M) communication systems based on the Third Generation Partnership Project (3GPP) Long Term Evolution (LTE), the machine type communication (MTC) devices compete in a random access channel (RACH) to access the network. An MTC device randomly chooses a preamble from a pool of preambles and transmits it during the RACH. The evolved node B (eNodeB) acknowledges the successful reception of a preamble if that preamble is transmitted by only one device. To reduce the burstiness of the connection requests in heavy traffic situations, access class barring (ACB) is proposed in the 3GPP standard. Using ACB, an MTC device postpones its request in a RACH with a probability p. In this paper, we propose a new adaptive ACB scheme for congestion control of bursty M2M traffic. The optimal value of the ACB depends on the total number of MTC devices competing in a RACH. To estimate this number, we derive a joint conditional probability distribution function (PDF) for the number of preambles selected by zero or one MTC device, conditioned on the number of MTC devices that passed the ACB check. We design a maximum likelihood estimator using this PDF. We use this estimation to dynamically adjust the ACB factor. To further improve our estimation, we use Kalman filtering based on the dynamics of the system. Numerical results show that the total service time for the proposed method is very close to the optimal case where the information of the number of MTC devices is given.**

***Index Terms*—Machine type communication, congestion control, access class barring, Kalman filtering.**

I. Introduction

Machine to machine (M2M) communication is an enabling communication technology which facilitates the realization of the Internet of Things [1]. An M2M communication system consists of a large number of machine type communication (MTC) devices which can communicate with the MTC servers or other MTC devices to accomplish specific tasks. It is reported in [2] that by 2020, there will be 12.5 billion MTC devices in the world. Due to the significant increase in demand for M2M applications [3], several associations including Third Generation Partnership Project (3GPP) and the Institute of Electrical and Electronics Engineers (IEEE) have started standardization in this area [4].

M2M communications have several applications including monitoring of vital signs in health care system, monitoring of the oil pipelines, and smart metering [5]. The cellular networks can provide a suitable infrastructure for M2M communications. Cellular networks provide ubiquitous connection in most urban and rural areas. Therefore, there is no need to deploy new base stations dedicated to M2M communication [6]. The resources of the cellular network can be divided between human to human (H2H) communications and M2M communications.

Since cellular networks are mainly designed for H2H communications, we need to revisit their use for M2M communications. According to [5] and [7], the number of MTC devices within a cell can exceed tens of thousands of devices. For event-driven M2M applications, several MTC devices may be activated simultaneously. If all the activated devices try to access the base station or evolved node B (eNodeB) within a short interval, congestion would occur at the radio access network (RAN) [8].

To request an uplink connection, an MTC device randomly chooses and transmits a preamble during a logical shared uplink channel called random access channel (RACH). The eNodeB can distinguish a request if only one device has selected this preamble. Several approaches have been proposed to alleviate the congestion caused by the event-driven applications. In [9], [10], access class barring (ACB) mechanism has been proposed for RAN overload control. The ACB factor allows MTC devices to transmit their connection requests with different probabilities. Numerical results show the performance of using a fixed ACB factor in [11], [12]. In [13], [14], it is proposed to separate the RACHs used by H2H and M2M communications to avoid H2H users of being blocked from accessing the network in the presence of bursty M2M traffic.

In [15], the ACB factor and the timing advance are jointly used to reduce the RAN congestion. In [16], a heuristic algorithm is proposed to adaptively update the ACB factor. This update is based on the number of successful transmissions and the number of collisions in each time slot. This algorithm uses the fact that by increasing the ACB factor p, the number of transmitted preambles and the number of collisions are increased. On this basis, if the number of the observed collisions (i.e., simultaneous preamble transmissions) is more than a threshold, then the algorithm will decrease the ACB factor p for a certain value. On the other hand, if the number of successful transmissions is more than a threshold, then the ACB factor p will be increased.

In this paper, we investigate the problem of the overload control in RAN. We aim to reduce the total service time in the overload condition instead of rejecting the access in the core network. To achieve this goal, we determine an accurate estimate for the number of the MTC devices by using all the available information at eNodeB (i.e., the number of successful transmissions and the number of unused preambles in each time slot). This estimation is used to adaptively adjust the ACB factor based on the traffic condition. The main contributions of this paper can be summarized as follows:

- We first derive a new joint conditional probability distribution function (PDF) for the number of preambles which are selected by zero, one, and multiple MTC devices conditioned on the number of MTC devices that have passed the ACB check.
- We propose a maximum likelihood estimator using this PDF to estimate the total number of the MTC devices in the system. This estimation is used to adaptively adjust the ACB factor.
- To further refine the estimation, we propose to use Kalman filter which takes into account the dynamics of the system.
- The simulation results show that by estimating the number of MTC devices, the total service time can approach the optimal case where the information of the number of devices is given.

The rest of this paper is organized as follows. In Section II, the system model and the problem formulation are presented. In Section III, we use Kalman filter to further improve the estimation accuracy. Performance evaluation is presented in Section IV and Conclusions are given in Section V.

II. System Model and Problem Formulation

A. System Model

We consider N MTC devices registered with an eNodeB. We assume that these devices are activated and request a connection with the eNodeB within an interval $(0, T_{\mathrm{A}})$. This can be due to several reasons including an alarm or power shortage recovery. Each MTC device will be activated at time $t \in [0,\ T_{\mathrm{A}}]$ with probability $g(t)$. Two distributions have been proposed for $g(t)$ in [11]. One of them follows a beta distribution and the other one follows a uniform distribution in interval $[0,\ T_{\mathrm{A}}]$. In this paper, we assume that eNodeB, does not have any knowledge about $g(t)$.

The activation time is divided into time slots in the system. Each time slot starts at the beginning of a RACH and ends by the start of the next RACH. We use I_{A} to denote the number of time slots within the activation interval. Each time slot contains two parts. The first part is the uplink random access channel. The second part is used for downlink acknowledgement and data transmission. As illustrated in Fig. 1, the i^{th} time slot starts at t_{i-1} and ends at t_i. Using the PDF of $g(t)$, the average number of newly arrived MTC devices in the i^{th} time slot is

$$\lambda_i = E\left[a_i \mid N\right] = N \int_{t_{i-1}}^{t_i} g(t)\,\mathrm{d}t, \quad i = 1, 2, \ldots, I_{\mathrm{A}} \tag{1}$$

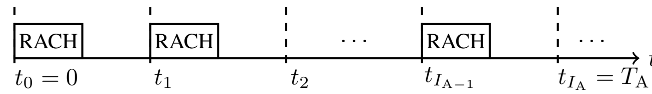

Fig. 1. Random access procedure in LTE networks.

where the random variable a_i is the number of new arrivals in the i^{th} time slot and it follows a binomial distribution.

Let r denote the number of the preambles assigned by the eNodeB for M2M communications. As a method to control congestion, at the beginning of a time slot and before each random access, eNodeB broadcasts an ACB factor. The ACB factor is a real number between zero and one. It can be changed from one time slot to another. We denote this number in the i^{th} time slot by p_i. After receiving the ACB factor at the beginning of time slot i, each MTC device which is activated generates a number between 0 and 1. Then it performs an ACB check by comparing the generated number with p_i. If this number is less than p_i, then a preamble is selected randomly and transmitted for that RACH. Otherwise, it will be backlogged until the next time slot.

Those MTC devices that have passed the ACB check can independently select one of the r preambles. If a preamble is selected by more than one user, collision will occur. The eNodeB is not able to decode a collided packet. Thus, none of those devices can access the channel and they have to try in the next time slots again. In time slot i, the number of successful transmissions is denoted by s_i. We know that after time T_A (i.e., $i > I_A$), no more devices activates. The dynamics of the system can be described as follows

$$N_{i+1} = \begin{cases} N_i - s_i + a_i & \text{if } i \le I_{\mathrm{A}} \\ N_i - s_i & \text{if } i > I_{\mathrm{A}} \end{cases}, \tag{2}$$

where N_i is the sum of the backlogged users and new arrivals at the beginning of the i^{th} time slot.

According to [16], in time slot i, the optimal value of the ACB factor is equal to

$$p_i^* = \min\left\{1, \frac{r}{N_i}\right\}. \tag{3}$$

The eNodeB does not have the information of the value of N_i. Therefore, we estimate this value using the information available at eNodeB and the dynamics of the system. In the next subsection, we focus on estimating the value of N_i.

B. Problem Formulation

In time slot i and after the RACH, the available information at the eNodeB is the number of unused preambles u_i and the number of successful preamble transmissions s_i. Let n_i denote the number of MTC devices that have passed ACB check. We derive the probability of observing u_i unused preambles and s_i successful preamble transmissions conditioned on n_i.

Let A_j denote the event that preamble j is selected by at most one MTC device. Then, $P(A_j)$ is equal to the probability that preamble j is selected by exactly one MTC device plus the probability that it is not selected by any MTC devices. This probability can be written as

$$P(A_j) = \left(1-\frac{1}{r}\right)^{n_i} + \binom{n_i}{1}\frac{1}{r}\left(1-\frac{1}{r}\right)^{n_i-1}. \quad (4)$$

Let π_k denote the joint probability that k specific preambles are selected by no more than one MTC device. π_k can be written as

$$\pi_k = \sum_{l=0}^{k}\binom{k}{l}\binom{n_i}{l}\frac{l!}{r^l}\left(1-\frac{k}{r}\right)^{n_i-l}, \; k=1,\ldots,r. \quad (5)$$

Here, the summation variable l can be interpreted as the number of preambles that are selected by exactly one MTC device. Since we are interested in the probability of selecting k specific preambles, for each l, there are $\binom{k}{l}$ ways to choose the preambles and l MTC devices can be chosen in $\binom{n_i}{l}$ ways. Assuming the devices as various balls and the preambles as different baskets, these devices can be placed in the selected preambles in $l!$ different ways. The remaining $n_i - l$ MTC devices are distributed among $r-k$ preambles in $(r-k)^{n_i-l}$ different ways. Dividing these multiplications by r^{n_i} gives the probability π_k. We notice that due to the symmetry of the problem, π_k is independent of the set of preambles (i.e., it is valid for any set of k-preambles). The probability that any k preambles are selected by no more than one MTC device, denoted by S_k, can be computed as

$$\begin{aligned} S_k &= \binom{r}{k}\pi_k \\ &= \binom{r}{k}\sum_{l=0}^{k}\binom{k}{l}\binom{n_i}{l}\frac{l!}{r^l}\left(1-\frac{k}{r}\right)^{n_i-l}. \end{aligned} \quad (6)$$

Using the inclusion and exclusion principle [17] along with equation (6), we can calculate the probability of $P\left(\bigcup_{j=1}^{r} A_j\right)$ as

$$\begin{aligned} P\Big(\bigcup_{j=1}^{r} A_j\Big) &= \sum_{k=1}^{r}(-1)^{k+1}S_k \\ &= \sum_{k=1}^{r}\sum_{l=0}^{k}(-1)^{k+1}\binom{r}{k}\binom{k}{l}\binom{n_i}{l}\frac{l!}{r^l}\left(1-\frac{k}{r}\right)^{n_i-l}, \end{aligned} \quad (7)$$

where $P\left(\bigcup_{j=1}^{r} A_j\right)$ is the probability that at least one preamble is selected by at most one MTC device. Let $P_2(n_i, r)$ denote the probability that each preamble is selected by at least two MTC devices. This is a complementary event to $\left(\bigcup_{j=1}^{r} A_j\right)$ and its probability can be calculated as

$$\begin{aligned} P_2(n_i, r) &= 1 - P\Big(\bigcup_{j=1}^{r} A_j\Big) \\ &= 1 - \sum_{k=1}^{r}\sum_{l=0}^{k}(-1)^{k+1}\binom{r}{k}\binom{k}{l}\binom{n_i}{l}\frac{l!}{r^l}\left(1-\frac{k}{r}\right)^{n_i-l} \\ &= \sum_{k=0}^{r}\sum_{l=0}^{k}(-1)^{k}\binom{r}{k}\binom{k}{l}\binom{n_i}{l}\frac{l!}{r^l}\left(1-\frac{k}{r}\right)^{n_i-l}. \end{aligned} \quad (8)$$

The final step is to derive the joint conditional probability that u_i preambles are unused and s_i preambles are selected by exactly one MTC device. There are $\binom{r}{u_i+s_i}$ ways to choose $u_i + s_i$ preambles out of r preambles. From these preambles, there are $\binom{u_i+s_i}{s_i}$ ways to choose s_i preambles. Moreover, the s_i MTC devices can be chosen in $\binom{n_i}{s_i}$ different ways from n_i devices. These devices can be replaced in the selected preambles in $s_i!$ ways. The remaining $n_i - s_i$ MTC devices are distributed among $r - u_i - s_i$ preambles, such that each preamble is selected by more than one MTC device. This can be done in $(r-u_i-s_i)^{n_i-s_i}P_2(n_i-s_i, r-u_i-s_i)$ different ways. Dividing these multiplications by r^{n_i} results in

$$\begin{aligned} P(u_i, s_i \mid n_i) = &\binom{r}{u_i+s_i}\binom{u_i+s_i}{s_i}\binom{n_i}{s_i}s_i! \\ &\times \frac{(r-u_i-s_i)^{n_i-s_i}}{r^{n_i}}P_2(n_i-s_i, r-u_i-s_i). \end{aligned} \quad (9)$$

Now, we can estimate n_i by using maximum likelihood estimator

$$\hat{n}_i = \arg\max_{n_i} P(u_i, s_i \mid n_i). \quad (10)$$

In an M2M system based on 3GPP LTE with r preambles, u_i and s_i are known to the eNodeB after performing a RACH. To estimate the number of devices using the estimator in (10), one can use Newton method. We employ another method to reduce the complexity during its online operation. There is a unique corresponding maximum likelihood estimation for each observed u_i and s_i. Thus, this problem can be solved offline numerically. A lookup table is obtained with r rows and r columns for different values of u_i and s_i, respectively. We denote this table by T. We notice that this table is computed once and it is used throughout the operation of the system. $\hat{n}_i$ can be calculated using this table as

$$\hat{n}_i = T(u_i, s_i). \quad (11)$$

Next, we estimate N_i by using $\hat{n}_i$ and the ACB factor p_i. The value of $\hat{n}_i$ is known from (11). We know that N_i is a random variable which follows a negative binomial distribution with parameters n_i and p_i. We estimate this distribution by replacing n_i with its approximate value $\hat{n}_i$. Thus, we can calculate the mean square estimation of N_i, which is the

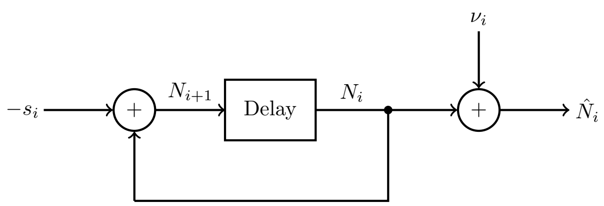

Fig. 2. A dynamic linear system.

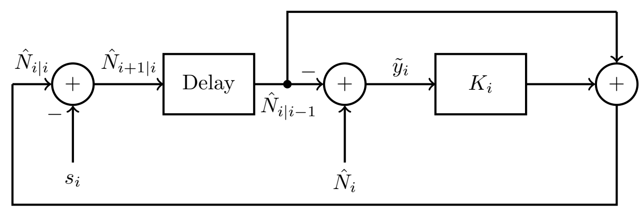

Fig. 3. Structure of the Kalman filter.

expectation of the random variable N_i conditioned on our observation $\hat{n}_i$ as

$$\hat{N}_i \triangleq E[N_i \mid \hat{n}_i] = \frac{\hat{n}_i}{p_i} = \frac{T(u_i, s_i)}{p_i}. \quad (12)$$

Moreover, the variance of N_i can be calculated as

$$\mathrm{Var}[N_i \mid \hat{n}_i] = \frac{\hat{n}_i(1-p_i)}{p_i^2} = \frac{(1-p_i)}{p_i^2}T(u_i, s_i). \quad (13)$$

We will use this variance in the next section. Observing u_i and s_i at time slot i, one can set the ACB factor p_{i+1} for time slot $i+1$ adaptively using the equations (2) and (12) along with the fact that $p_i^* = \min\{1, \frac{r}{N_i}\}$. We have

$$\hat{p}_{i+1} = \begin{cases} \min\{1, \frac{r}{\hat{N}_i - s_i + E[a_i]}\} & \text{if } i \leq I_\text{A} \\ \min\{1, \frac{r}{\hat{N}_i - s_i}\} & \text{if } i > I_\text{A} \end{cases}. \quad (14)$$

Due to the fact that $g(t)$ is unknown at eNodeB, it does not have the value of $E[a_i]$. However, we know that the time between two consecutive RACHs is not significant. Assuming that variations in $g(t)$ is negligible between two consecutive RACHs, one can approximate $E[a_i]$ by $E[a_{i-1}]$. An estimate for $E[a_{i-1}]$ can be obtained using equation (2) along with the fact that $a_{i-1} \geq 0$ as $\max\{0, \hat{N}_i - \hat{N}_{i-1} + s_{i-1}\}$. Therefore, we have

$$E[a_i] \approx \max\{0, \hat{N}_i - \hat{N}_{i-1} + s_{i-1}\}. \quad (15)$$

Thus, $\hat{p}_{i+1}$ can be calculated as

$$\hat{p}_{i+1} = \begin{cases} \min\{1, \frac{r}{\hat{N}_i - s_i - \max\{0, \hat{N}_i - \hat{N}_{i-1} + s_{i-1}\}}\} & \text{if } i \leq I_\text{A} \\ \min\{1, \frac{r}{\hat{N}_i - s_i}\} & \text{if } i > I_\text{A} \end{cases}. \quad (16)$$

Note that for the first time slot, we can compute the optimal ACB by using (3). Moreover, we have $\hat{N}_0 = 0$ and $s_0 = 0$.

III. Further Improvement By Using Kalman Filter

In the previous section, we only use the information of each time slot at eNodeB to estimate the ACB factor. In this section, we take into account the correlation between N_{i+1} and N_i in addition to the information about the number of preambles selected by zero and one MTC device. We estimate the state of the linear system N_i by using Kalman filter [18]. Since the distribution of a_i is unknown at eNodeB, we only use this estimation for the time slots greater than I_A, where there is no more incoming traffic. The dynamics of the system, which is illustrated in Fig. 2, can be modeled as

$$N_{i+1} = N_i - s_i, \quad (17)$$

$$\hat{N}_i = N_i + \nu_i, \quad (18)$$

where ν_i is the estimation error. In (18), $\hat{N}_i$ is the estimation obtained from (12). In order to use Kalman filter, it is necessary that ν_i are zero mean independent, identically distributed (i.i.d.) Gaussian random variables for all $i \geq I_\text{A}$. N_i follows a negative binomial distribution with mean $\hat{N}_i$ and variance $\mathrm{Var}[N_i \mid \hat{n}_i]$. The negative binomial distribution can be approximated as a normal distribution under special circumstances. Then, random variables ν_i for all i are independent, zero mean Gaussian random variables with variance $\mathrm{Var}[N_i \mid \hat{n}_i]$ given in (13).

We employ the Kalman filter to find an estimation of N_i based on the measurements up to time slot i. Let $\underline{Z}_i \triangleq \{\hat{N}_{I_\text{A}}, \hat{N}_{I_\text{A}+1}, \ldots, \hat{N}_i\}$, then the *a priori* estimation of N_{i+1} is defined as $\hat{N}_{i+1|i} = E[N_{i+1} \mid \underline{Z}_i]$, which is the maximum likelihood estimation. Here, the term *a priori* means that the estimator has used the observations before time slot $i+1$. The *a posteriori* estimate of N_{i+1} is also defined as $\hat{N}_{i+1|i+1} = E[N_{i+1} \mid \underline{Z}_{i+1}]$. The Kalman filter is used to derive these two estimations. According to [18], the Kalman filter has two distinct phases, namely *predict* and *update*. In the predict phase, the *a posteriori* estimate of the previous time slot is used to provide *a priori* state estimate $\hat{N}_{i+1|i}$ as

$$\hat{N}_{i+1|i} = \hat{N}_{i|i} - s_i, \quad (19)$$

$$P_{i+1|i} = P_{i|i}, \quad (20)$$

where in (20), $P_{i+1 \mid i}$ and $P_{i \mid i}$ are *a priori* and *a posteriori* error covariances, respectively. They are defined as

$$P_{i+1|i} \triangleq E[(N_{i+1} - \hat{N}_{i+1|i})^2 \mid \underline{Z}_i], \quad (21)$$

$$P_{i|i} \triangleq E[(N_i - \hat{N}_{i|i})^2 \mid \underline{Z}_i]. \quad (22)$$

In the *update* phase, the *a priori* estimate is combined with the current observation to refine the estimation. This improved estimate is called *a posteriori* state estimate. It is denoted by

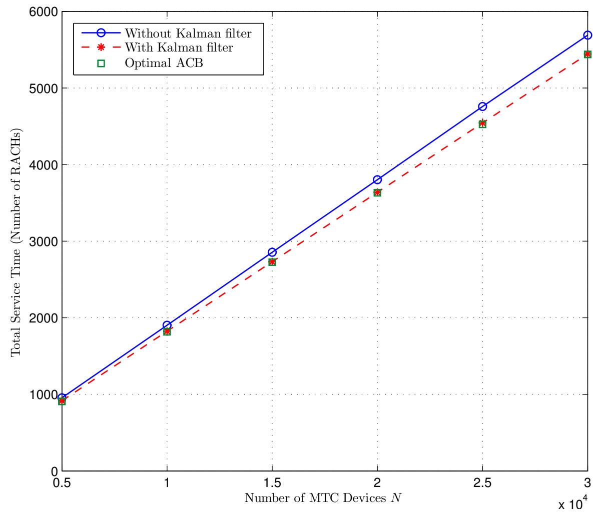

Fig. 4. The total service time vs the number of MTC devices N with $r = 15$ and $I_{\mathrm{A}} = 100$.

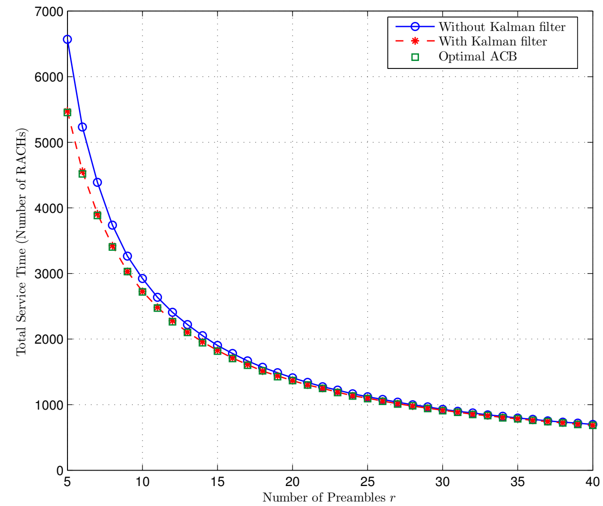

Fig. 5. The total service time vs the number of preambles r with $N = 10000$ and $I_{\mathrm{A}} = 100$.

$\hat{N}_{i+1|i+1}$ and can be calculated as

$$\tilde{y}_{i+1} = \hat{N}_{i+1} - \hat{N}_{i+1|i}, \tag{23}$$

$$K_{i+1} = \frac{P_{i+1|i}}{P_{i+1|i} + R_{i+1}}, \tag{24}$$

$$\hat{N}_{i+1|i+1} = \hat{N}_{i+1|i} + K_{i+1}\tilde{y}_{i+1}, \tag{25}$$

$$P_{i+1|i+1} = (1 - K_{i+1})P_{i+1|i}, \tag{26}$$

where $\tilde{y}_{i+1}$ is the measurement residual, which is the difference of $\hat{N}_{i+1}$ and $\hat{N}_{i+1|i}$. Moreover, K_{i+1} is the optimal Kalman gain. In (24), R_{i+1} is the covariance of the observation noise ν_{i+1}, which is equal to $\mathrm{Var}[N_{i+1} \mid \hat{n}_{i+1}]$ and can be calculated from (13). The structure of Kalman filter used in our system is depicted in Fig. 3. As mentioned earlier, we only employ Kalman filter in time slots after I_{A}. Therefore, we can use $\hat{N}_{I_{\mathrm{A}}}$ as an initialization for Kalman filter. We can also initialize $P_{I_{\mathrm{A}}|I_{\mathrm{A}}}$ by $\mathrm{Var}[N_{I_{\mathrm{A}}} \mid \hat{n}_{I_{\mathrm{A}}}]$.

The estimated value for the optimal ACB factor at the beginning of time slot $i + 1$ for $i \geq I_{\mathrm{A}}$ can be calculated using equation (3) as

$$\hat{p}_{i+1} = \min\left\{1, \frac{r}{\hat{N}_{i+1|i}}\right\}. \tag{27}$$

Note that the eNodeB should be able to determine the ACB factor p_{i+1} before the beginning of time slot $i+1$. That is the reason the *a priori* estimate $\hat{N}_{i+1|i}$ is used in (27).

IV. Performance Evaluation

In this section, the proposed methods with and without using Kalman filter are compared with the case optimal ACB is employed. To obtain the optimal ACB curve, we assume that the eNodeB is aware of the total number of the active MTC devices in the system and sets the ACB factor using (3). We notice that in reality, such information is not available at eNodeB. In our simulations, we measure the total service time. It is the mean number of the required RACHs to finish

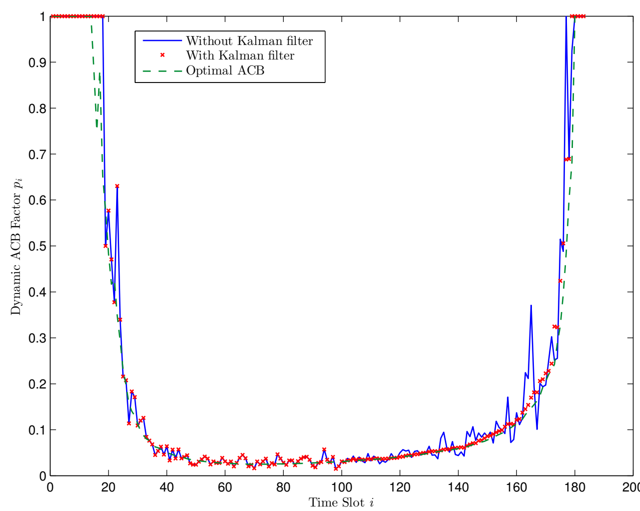

Fig. 6. Dynamic ACB factor p vs time slot with $N = 1000$, $r = 15$ and $I_{\mathrm{A}} = 100$.

all the transmissions. In our simulations, first, for a given number of preambles, the total service time versus the total number of MTC devices activated in the activation interval is plotted. Then, for a given number of MTC devices, we plot the total service time versus the number of preambles. In the simulations, we assume that each MTC device is activated under beta distribution with parameters $\alpha = 3$, $\beta = 4$ as [5]

$$g(t) = \frac{t^{\alpha-1}(T_{\mathrm{A}} - t)^{\beta-1}}{T_{\mathrm{A}}^{\alpha+\beta-1}\mathcal{B}(\alpha, \beta)}, \tag{28}$$

where $\mathcal{B}(\alpha, \beta)$ is the beta function [19].

For the first simulation, the number of RACHs within the activation time and the number of preambles are $I_{\mathrm{A}} = 100$ and $r = 15$, respectively. We use $I_{\mathrm{A}} = 100$ for all the simulations. $r = 15$ is a suggested value for M2M systems [5]. Fig. 4 shows the total service time as the number of the MTC devices varies from 5000 up to 30000. It can be seen that the method with Kalman filter performs close to

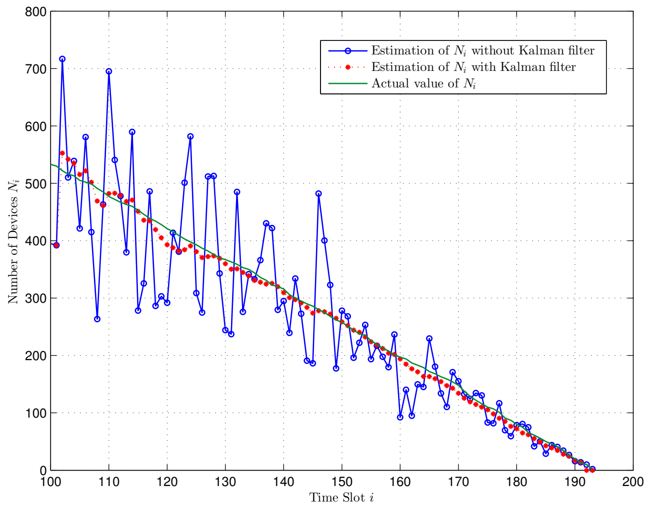

Fig. 7. Number of backlogged MTC devices N_i vs time slot with $N = 1000$, $r = 15$ and $I_{\rm A} = 100$.

the optimal curve. When Kalman filter is not employed, the eNodeB estimates the number of MTC devices and adjusts the ACB factor accordingly. Since the Kalman filter uses not only the information in the current time slot but also the information from the previous time slots, one can expect that it has better performance compared to the case Kalman filter is not employed.

For the second simulation depicted in Fig. 5, the total number of MTC devices N is equal to 10000 and the number of assigned preambles for M2M communication varies from 5 to 40. As we expect, the total service time decreases as the number of the preambles increases. This figure confirms the observation that the proposed scheme performs close to the optimal ACB case. In Fig. 6, the variation of ACB p_i over time is presented while we run the simulation once. In this figure, the total number of MTC devices and the number of preambles are $N = 1000$ and $r = 15$, respectively. For time slots $i \leq I_{\rm A} = 100$, the ACB factor of the proposed method with Kalman filter is the same as the method without Kalman filter. However, for the subsequent time slots, the ACB factor obtained by employing Kalman filter is very close to the optimal ACB with smooth variations. Intuitively, this results from the fact that Kalman filter prevents rapid changes in the ACB factor by taking into account the correlation between different time slots. The variations of the ACB factor for the scheme without Kalman filter is high in some cases however.

Fig. 7 shows the variation of the number of MTC devices N_i versus time slot with the same parameters used to obtain Fig. 6. Since we use Kalman filter for the time slots greater than I_A, Fig. 7 only includes time slots $i > I_A$. In this figure, for the method without Kalman filter, the variation of the estimated N_i is significant. However, in the method with Kalman filter, the variation of this estimation is smooth near its actual value.

V. Conclusion

In M2M systems, for the cases that the activation time of the MTC devices has a bursty pattern, the RAN congestion is inevitable. In this paper, we presented a new overload control scheme for bursty M2M traffic in LTE networks. In the proposed scheme, we used both the slot information and the dynamics of the system to estimate the total number of the backlogged MTC devices in each time slot. We first derived the joint conditional PDF for the number of unused preambles and the number of successful transmissions conditioned on the number of MTC devices that have passed the ACB check. Then, we estimated this number by using a maximum likelihood estimator. We used this estimation to adjust the ACB factor dynamically. Finally, we used Kalman filter to further improve the estimation accuracy. Simulation results show that the proposed method performs very close to the optimal case. For future work, we will consider dynamic preamble assignment. Another direction is to modify the approach to consider delay sensitive and delay tolerant applications separately.

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 communication in consumer electronics 2012-22 technical report," Feb. 2013.

[3] Cisco, San Jose, CA, USA, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019 technical report," Feb. 2015.

[4] IEEE, "Machine to Machine (M2M) communications technical report," *IEEE 802.16p-10/0005*, Nov. 2010.

[5] 3GPP, "Study on RAN Improvements for machine-type communications," 3rd Generation Partnership Project (3GPP), TR 37.868 V11.0.0, Oct. 2011.

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

[7] Huawei and CATR, "R2-100204: Traffic model for M2M services," presented at the 3GPP TSG RAN WG2 Meeting #68bis, Jan. 2010.

[8] 3GPP, "Medium access control (MAC) protocol specification," 3GPP TS 36.321 V12.4.0, Dec. 2014.

[9] 3GPP, "Access class barring and overload protection," 3GPP TR 23.898 V7.0.0, Mar. 2005.

[10] C.M. Chou and C.Y. Huang, "Loading prediction and barring controls for machine type communication," in *Proc. of IEEE ICC*, Budapest, Hungary, June 2013.

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

[12] 3GPP, "MTC simulation results with specific solutions," TSG RAN WG2 #71 R2-104662, Aug. 2010.

[13] J.P. Cheng, C.H. Lee, and T.M. Lin, "Prioritized random access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *Proc. of IEEE GLOBECOM,* Houston, TX, Dec. 2011.

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

[15] Z. Wang and V.W.S. Wong, "Joint access class barring and timing advance model for machine-type communications," in *Proc. of IEEE ICC*, Sydney, Australia, June 2014.

[16] S. Duan, V. Shah-Mansouri, and V.W.S. Wong, "Dynamic access class barring for M2M communications in LTE networks," in *Proc. of IEEE GLOBECOM*, Altlanta, GA, Dec. 2013.

[17] W. Feller, *An Introduction to Probability Theory and Its Application*, John Wiley and Sons, 1968.

[18] B.D. O. Anderson and J.B. Moore, *Optimal Filtering*, Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

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