Multiple Access Class Barring factors Algorithm for M2M communications in LTE-Advanced Networks

Meriam Bouzouita
University of Rennes, Rennes, France
SUP'COM, Ariana, Tunisia
mariem.bouzouita@supcom.tn

Yassine Hadjadj-Aoul
University of Rennes, Rennes, France
yhadjadj@irisa.fr

Nawel Zangar
SUP'COM, Tunisia
nawel.zangar@insat.rnu.tn

Gerardo Rubino
INRIA, Rennes
Gerardo.Rubino@inria.fr

Sami Tabbane
SUP'COM, Tunisia
Sami.Tabbane@insat.rnu.tn

ABSTRACT

The forecast dramatic growth, of the number of Machine-to-Machine (M2M) communications, challenges the traditional networks of Mobile Network Operators (MNO). In fact, a large number of devices may attempt simultaneously to access the base station, which may result in severe congestions at the random-access channel (RACH) level. To alleviate such congestion while regulating the M2M devices' opportunities to transmit, the Access Class Barring (ACB) process was proposed. In this article, we propose a novel implementation of the ACB mechanism in the context of multiple M2M traffic classes. Based on a scheduling algorithm, we have applied a PID controller to adjust dynamically multiple ACB factors related to each class category, guaranteeing a number of devices around an optimal value that maximizes the Random Access (RA) success probability. The obtained results demonstrate the efficiency of the proposed mechanism by increasing the success probability and minimizing radio resources' underutilization with respect to each class priority.

Keywords

M2M, MTC, Access Class Barring, Congestion, Random Access.

1. INTRODUCTION

Machine Type Communications (MTC) or Machine-to-Machine (M2M) communications are nowadays gaining a huge interest from the stakeholders, and particularly the Mobile Network Operators (MNO), and their customers. In fact, M2M communications are seen as one of the most important opportunities to face the revenue's cuts for mobile operators while providing a plethora of services to the customers.

These services can be declined in a wide range of automated applications covering a large number of domains [1]. These applications can be classified based on their intrinsic characteristics (e.g. event driven) and the corresponding traffic requirements. For instance, we can identify the most important MTC scenarios such as: surveillance and remote monitoring, smart metering, emergency use cases, fleet management, etc [2].

The huge number of M2M devices, which may attempt, at the same time, to access the base station, may result in severe congestions at the random-access channel (RACH) level. In fact, a large number of devices may be triggered simultaneously and attempt to perform the Random Access (RA) in order to request for uplink radio resources. This congestion is even more aggravated when considering the class of event-driven communications, in which a large number of devices is activated during a very short period of time. These devices contend for a limited number of resources, called preambles. Indeed, if two or more MTC equipments choose the same preamble, the Evolved Node B (eNB) will be unable to identify the initiator of the RA and a collision will happen [3]. This may reduce the success access probability and may result in a performance degradation for MNO.

In this paper, our main concern is to design an efficient mechanism to maximize the wireless resources' utilization while guaranteeing the access priorities that may exist between the different class of applications. Another important concern consists in protecting the M2M event-driven communications by prioritizing their access to the channel while adapting rapidly the ACB factors to absorb this type of traffic.

To achieve this objective, we proposed both: (1) a Proportional Integral Derivative (PID) controller to make the number of M2M devices attempting the access procedure converge to the optimal one, and (2) a scheduling algorithm to ensure a weighted proportional fairness among M2M devices of the different classes except the class of event-driven devices, which is prioritized.

The remainder of this article is organized as follows. Section 2 provides an overview of the ACB mechanism. Section 3 is dedicated to the description of the proposed mechanism.

Section 4 portrays the simulation setup and discusses the obtained results. Finally, the paper concludes in Section 5 with a summary recapping the main advantages and achievements of the proposed scheme.

2. BACKGROUND

To tackle the problem of RACH congestion, many overload resolution methods were proposed in the literature [4]. The ACB mechanism, which can be used in conjunction with other approaches, is considered as one of the most efficient ones, as it handles the problem at its root. Indeed, this mechanism allows resources' saving by dissuading some M2M devices from attempting to perform the RA.

The main idea behind the ACB mechanism, in LTE-Advanced networks, is to broadcast a barring factor *ac-BarringFactor* and a barring time *ac-BarringTime*, which are used by an M2M device to determine whether it is temporarily barred from accessing the network.

The *ac-BarringFactor* determines the probability that a device is allowed or not to attempt to access the wireless resources. Before attempting the RA and establishing a connection, every device should first pass the ACB check.

Initially, each device generates a uniform random number q in the interval $[0, 1]$. If q is less than the ACB factor, the device starts the RA procedure; otherwise, the device will be blocked during a barring duration. Thus, if the value of this factor is properly set, this may help to ensure both small contentions on radio resources and low access delays for MTC devices.

Using the *ac-BarringTime*, each MTC device determines the blocking duration *Tbarring* before retrying a second ACB attempt after a first failed trial. This duration is calculated as follows:

$$\mathit{Tbarring} = (0.7 + 0.6 * q_b) * \mathit{ac\text{-}BarringTime}$$

where q_b represents a uniform random number in the interval $[0, 1]$ generated by the MTC device after a first failed ACB check [5].

The whole ACB process executed by the M2M devices is illustrated in Fig. 1.

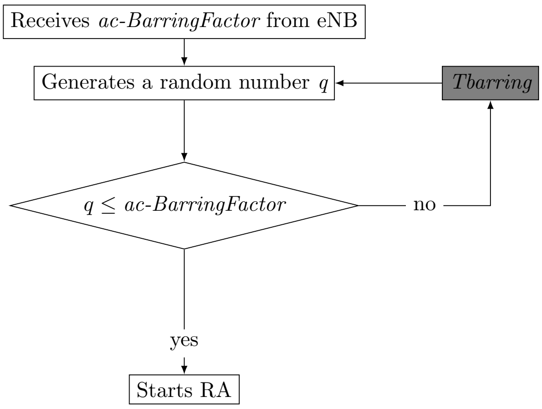
Receives ac-BarringFactor from eNB
Generates a random number q
Tbarring
q ≤ ac-BarringFactor
no
yes
Starts RA

Figure 1: ACB procedure

Many research works investigated the ACB scheme to reduce the congestion impacts. In [6], the authors proposed a cooperation among eNBs as a way to eliminate the substantial defects existing in the classical ACB. The scheme distributes efficiently the load between eNBs and improves significantly the access delays. However, in this approach, the number of devices that attach to the eNBs are not considered, which limits the efficiency of the mechanism in reducing the contention for a particular cell. The authors in [7] address this last limitation, by considering the number of MTC devices to determine the probability that an equipment may access the eNB, which allows obtaining a better set of ACB parameters than the one devised in [6]. In [8], a new heuristic algorithm was proposed in order to adaptively change the ACB factor without prerequisite knowledge of the number of backlogged users. The obtained results show that the proposed heuristic algorithm achieves good performance as it reduces significantly the access delays. However, this approach lacks responsiveness and is based on several parameters. The tuning of these could considerably influence the effectiveness of the approach.

We propose in this paper a new method adapting dynamically, in real-time, the ACB factor according to both the contention's level and the M2M classes.

3. MULTIPLE ACCESS CLASS BARRING ALGORITHM

M2M devices' scheduling is one of the most important functions, which not only ensures avoiding the congestion in such contention-based access networks, but also guarantees the access for prioritized applications and weighted fairness for the others. This is particularly important for some M2M applications (i.e. event-driven), as the huge number of devices, which attempt to connect during a very short period of time, increases significantly the risk of congestion collapse.

Many ACB-based algorithms were proposed in the literature. However, most of them do not consider multi-class M2M devices. Thus, the obtained ACB factor is applied to all classes of M2M devices, which might be harmful for MNO networks. The originality of our approach resides, first, in considering devices belonging to different classes. Second, in opposition to existing works, we propose to find out the optimal number of devices and devising, then, an efficient controller to make the number of devices attempting the RA converge to the optimal one.

3.1 System Model

In this paper, we consider the classical architecture described in [1]. We propose, in the following, to model the RA process, as described in [3].

Our model for M2M devices' random access with multiple ACB factors (i.e. one factor per class) is illustrated in the following diagram. The model is a fluid one: the involved quantities, the whole numbers, are seen here as real quantities. The parameters used in the proposed system model are listed below:

- $x_{1,i}(t)$: the number of backlogged MTC devices from class i at time t, where $i \in \{1, 2, \ldots, k\}$. The constant

k represents the number of considered classes.

- $x_2(t)$: the number of MTC devices that pass the ACB check and wait to start RA attempt at time t.
- $x_3(t)$: the number of MTC devices that succeed RA procedure at time t.
- λ_i: the arrival rate of MTC devices from class $i \in \{1, 2, \ldots, k\}$. Different traffic patterns will be considered in the following, depending on the type of M2M application.
- $\theta_{1,i}(x_{1,i})$: the rate of ACB failure for class $i \in \{1, 2, \ldots, k\}$.
- $\theta_2(x_2)$: the rate of RA failures (i.e. collision and retransmission).
- $\mu(x_3)$: the rate of MTC departure after performing the RA successfully.
- $p_i(x_{1,i})$: the ACB factor for class $i \in \{1, 2, \ldots, k\}$.
- N: indicates the total number of radio resources (preambles) available during one time slot. It is a constant value.
- $R_s(x_2)$: denotes the number of MTC devices that transmitted their preambles successfully.

For the sake of clarity, we consider, in the figure below, only three classes of M2M devices (i.e. $k = 3$).

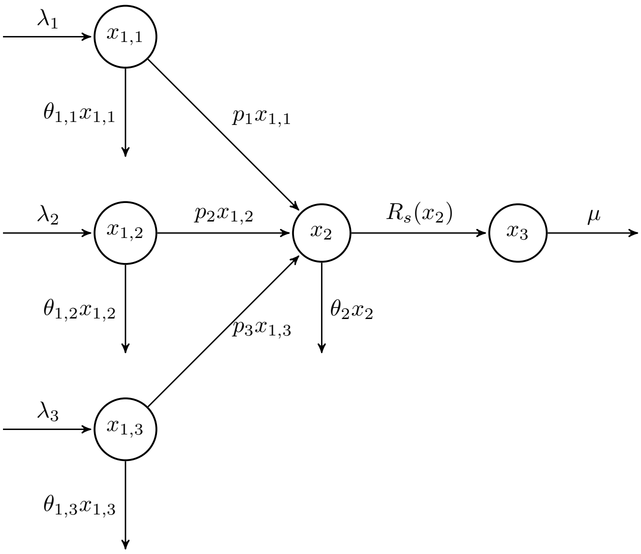

Figure 2: System description

Note that the system model illustrated above is influenced by the single-class system model proposed in [9].

Now we are ready to describe the evolution of the states $x_{1,i}$ with $i \in \{1, 2, \ldots, k\}$, x_2 and x_3 based on the model described in Fig. 2. Let first define $\mathbb{C} = \{1, 2, \ldots, k\}$ as the set of classes. The system's dynamics is described by the following system of differential equations:

$$\begin{cases} \dfrac{\mathrm{d}x_{1,i}}{\mathrm{d}t} = \lambda_i - p_i x_{1,i} - \theta_{1,i}x_{1,i}, & \text{for all } i \in \mathbb{C}, \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = \displaystyle\sum_{i=1}^{k} p_i x_{1,i} - R_s(x_2) - \theta_2 x_2, & \\ \dfrac{\mathrm{d}x_3}{\mathrm{d}t} = R_s(x_2) - \mu x_3, & \end{cases} \tag{1}$$

with the constraints that for all $i \in \mathbb{C}$, $x_{1,i}$, x_2 and x_3 should be nonnegative.

The function $R_s(x_2)$ is represented by the expected number of MTC devices succeeding in the access process. This number represents the number of preambles with only one device.

Let's define P_s as the probability that a given preamble is chosen by one MTC device (i.e. probability of success). To that purpose, we suppose that there are N available preambles in each RA opportunity and x_2 MTC devices contending for these resources. This is a typical "balls into bins' problem", in which P_s is given by:

$$P_s = \frac{x_2}{N}\left(1 - \frac{1}{N}\right)^{x_2 - 1}. \tag{2}$$

For a large N, this can be approximated by:

$$P_s = \frac{x_2}{N}\,\mathrm{e}^{-\dfrac{x_2 - 1}{N}}. \tag{3}$$

Therefore,

$$R_s(x_2) = x_2 e^{-\dfrac{x_2 - 1}{N}}. \tag{4}$$

The collision probability P_c can be derived using (3) and the idle probability (i.e. no user chooses a given preamble):

$$P_c = 1 - \frac{x_2}{N}\,\mathrm{e}^{-\dfrac{x_2 - 1}{N}} - e^{-\dfrac{x_2}{N}}. \tag{5}$$

To validate the proposed model, we will compare, in the performance and evaluation section, the analytical results of success and collision probabilities with those obtained by simulation.

3.2 Optimal number of M2M devices

The main idea, in this section, is to derive an optimal number of contending M2M devices x_2 in a way to be used by the proposed controller as an objective to achieve (i.e. targeted number of M2M devices).

All devices that have passed successfully the ACB test, will contend for the same radio resources (N) as a way to access the network. The optimal value of devices, performing RA

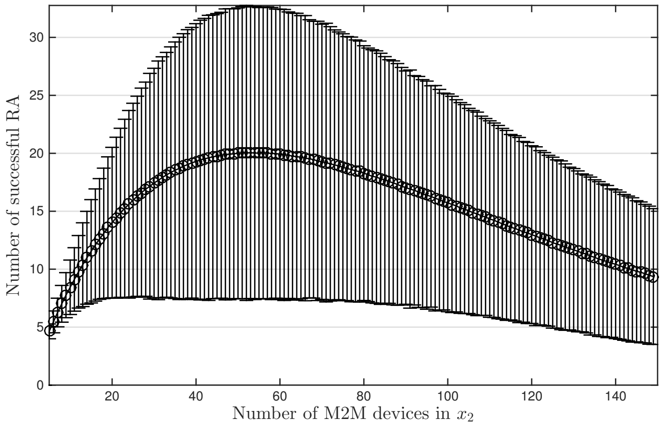

Figure 3: Successful RA

at the same time (i.e. x_2^{ref}) and maximizing the success probability, is obtained based on Monte-Carlo simulation.

For the simulations, we vary the number of M2M devices between 1 and 150 devices. We evaluated the average and the variance on the number of successful RA. To validate the obtained results, many seeds were tested and the results, obtained in Fig. 3, were similar.

The simulation results, depicted in Fig. 3, show that the maximum number of successful RA is obtained when the number of M2M devices in x_2 is equal to $(N-3)$. This number will be adopted as the optimal value (i.e. the targeted value) to generate the appropriate ACB factor in our scheduling algorithm.

3.3 PID feedback control on the access probability

In the proposed approach, the dynamic adjustment of the ACB factor (i.e. access probability) is achieved using a discrete Proportional Integral Derivative (PID) regulator [10]. The main idea, behind, is to make the total number of MTC devices, contending for the access, converges to the targeted value, which is defined in the previous subsection.

The discrete PID regulator can be described by the following equation [10]:

$$P_{acb}(n) = K_p e(n) + K_i \sum_{k=0}^{n} e(k) + K_d(e(n) - e(n-1)) \quad (6)$$

where n, P_{acb}, e, K_p, K_i and K_d represent respectively the instant, the controller output, the difference between the measured value and the set point value (i.e. the targeted value), the proportional gain, the integral gain and the derivative gain. In order to get the ideal response of the system, we considered the Ziegler-Nichols method [11] for the tuning of PID parameters.

As P_{acb} is a probability, a saturation block[1] is added to bind the values within the interval $[0,1]$.

The probability P_{acb}, calculated in (6), is general and do not concern a particular class of traffic. Thus, the probability P_{acb}^i for each class of service $i \in \mathbb{C}$ is derived from this value, as described in the next section. These values are broadcasted through a signaling channel to the different classes of M2M applications.

3.4 Proposed algorithm

In this subsection, we describe in details the proposed algorithm, which is illustrated in the figure below, to compute multiple ACB factors for the different classes of applications.

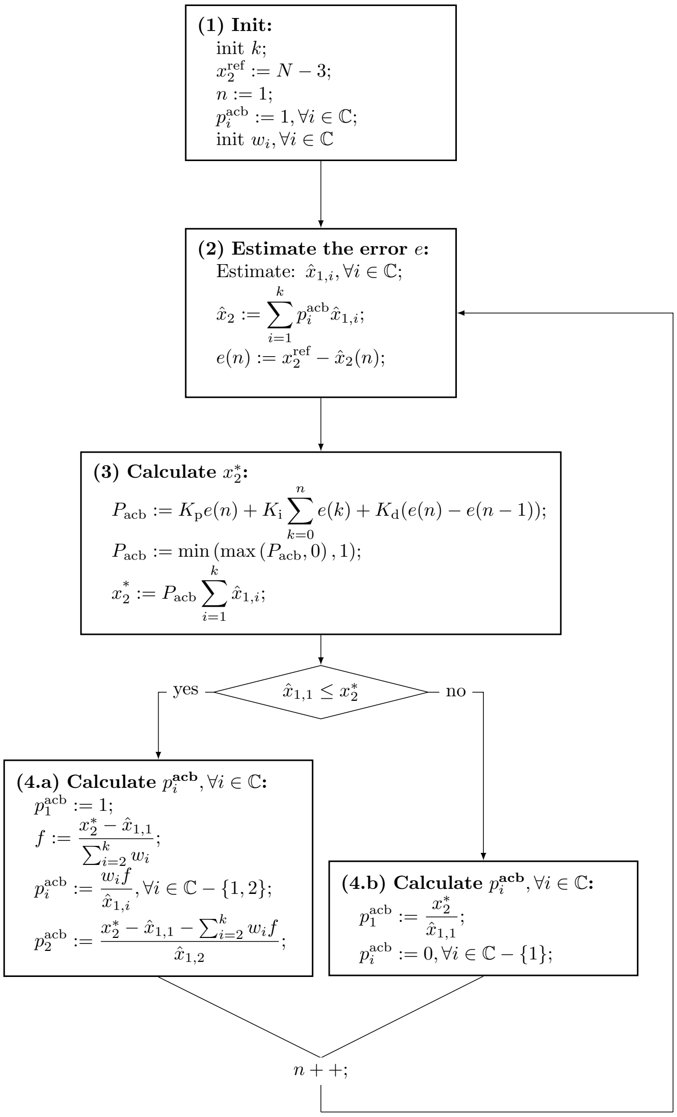
(1) Init:
init k;
$x_2^{\text{ref}} := N-3$;
$n := 1$;
$p_i^{\text{acb}} := 1, \forall i \in \mathbb{C}$;
init $w_i, \forall i \in \mathbb{C}$
(2) Estimate the error e:
Estimate: $\hat{x}_{1,i}, \forall i \in \mathbb{C}$;
$\hat{x}_2 := \sum_{i=1}^{k} p_i^{\text{acb}} \hat{x}_{1,i}$;
$e(n) := x_2^{\text{ref}} - \hat{x}_2(n)$;
(3) Calculate x_2^*:
$P_{\text{acb}} := K_{\text{p}} e(n) + K_{\text{i}} \sum_{k=0}^{n} e(k) + K_{\text{d}}(e(n) - e(n-1))$;
$P_{\text{acb}} := \min(\max(P_{\text{acb}}, 0), 1)$;
$x_2^* := P_{\text{acb}} \sum_{i=1}^{k} \hat{x}_{1,i}$;
yes
$\hat{x}_{1,1} \leq x_2^*$
no
(4.a) Calculate $p_i^{\text{acb}}, \forall i \in \mathbb{C}$:
$p_1^{\text{acb}} := 1$;
$f := \frac{x_2^* - \hat{x}_{1,1}}{\sum_{i=2}^{k} w_i}$;
$p_i^{\text{acb}} := \frac{w_i f}{\hat{x}_{1,i}}, \forall i \in \mathbb{C} - \{1,2\}$;
$p_2^{\text{acb}} := \frac{x_2^* - \hat{x}_{1,1} - \sum_{i=2}^{k} w_i f}{\hat{x}_{1,2}}$;
(4.b) Calculate $p_i^{\text{acb}}, \forall i \in \mathbb{C}$:
$p_1^{\text{acb}} := \frac{x_2^*}{\hat{x}_{1,1}}$;
$p_i^{\text{acb}} := 0, \forall i \in \mathbb{C} - \{1\}$;
$n++$;

Figure 4: Scheduling Algorithm

After a phase of initialization (at phase 1), the eNB estimates the error (i.e. e) to be used thereafter in the PID regulator (at phase 2). The error reflects the distance between the obtained value of the number of contending devices x_2 at step n and the targeted value x_2^{ref}. Therefore, the eNB estimates the number of successful ACB tests $\hat{x}_2$ by estimating the states $x_{1,i}$, for all $i \in \mathbb{C}$ (i.e. $\hat{x}_{1,i}$, for all $i \in \mathbb{C}$).

[1] A component imposing higher and lower bounds.

These estimates can be easily obtained by a per-class counting of the number of devices that have passed successfully the ACB test and by averaging these obtained values using an Exponentially Weighted Moving Average (EWMA).

After that (at phase 3), the eNB uses the PID controller to adjust the ACB factor (i.e. P_{acb}) enabling to converge towards the targeted number of contending devices (i.e. x_2^{ref}). As the output of the PID controller should be a probability, in spite of considering P_{acb}, we consider only values within the interval $[0, 1]$ by applying the: $\min(\max(P_{acb}, 0), 1)$. The computation of the blocking probability P_{acb} allows generating an optimal number of equipments that should pass successfully the ACB process (i.e. $\overset{*}{x_2}$) based on the estimated $\hat{x}_{1,i}$, for all $i \in \mathbb{C}$ (see phase 3).

In the proposed algorithm, we considered the first class as the most prioritized by using an absolute priority access, as it is the case for some M2M applications such as emergency services. The access probabilities of the devices belonging to other classes are balanced using a weight factor w_i, for all $i \in \mathbb{C}$. Depending on the availability of preambles for the first class, we pass by phase (4.a) or (4.b). Phase (4.b) is executed when there are not enough preambles to grant the access to all the devices of the first class (i.e. $\overset{*}{x_2}$ is smaller than $\hat{x}_{1,1}$). In this case the blocking probability for class 1 is calculated to have an optimal number of contending devices for this class $p_{acb}^1 = \frac{\overset{*}{x_2}}{\hat{x}_{1,1}}$. The devices from the other classes are blocked. When there are enough preambles to grant the access for all the devices in class 1, phase (4.a) is executed. In this case, p_{acb}^1 is equal to one (i.e. all the devices from the class are accepted), and the access is shared fairly between the other classes depending on the weight of each class (i.e. weighted fairness).

Once the ACB factors generation process is finished, it is broadcasted through a signaling channel to all the equipments, which should update their access probability and start ACB check. Then, the operation is repeated from phase 2, after incrementing the step variable n.

Note that the broadcasting of the access probabilities is repeated for each frame, which represents a delay of 10ms. Other delays might be considered for a less important accuracy and reactivity. The considered delay is short but represents a negligible overhead.

4. PERFORMANCE EVALUATION

4.1 Simulation parameters

Having described the details of the proposed algorithm to calculate a per-class ACB factor for heterogeneous M2M devices, we direct now our focus on evaluating its performance using the “Network Simulator (ns3)” environment [12].

The proposed model supports an unlimited and configurable number of types of M2M applications. However, for the simulations, we considered only the following classes:

- *Emergency applications*: this category of M2M applications includes prioritized services, triggered in emergency situations such as seismic alarms, which must be processed as fast as possible. Thus, this category requires the highest rate of successful accesses. It corresponds to class 1 in our model. As described in [13], emergency applications' arrival follows a Beta-based traffic model.
- *Applications for remote control and surveillance*: this category represents a class of M2M application with a good level of priority and continuous data transmission [14]. The model of arrival of this type of applications corresponds to a uniform process.
- *Smart-grid-related applications*: this category includes smart grid metering, water metering, gas metering, etc. A large number of devices with a reasonably large amount of data is periodically transferred to eNB and, thus, causing RAN overload. Such applications have a very low priority of access with delay-tolerance and can be rejected in case of congestion. The periodic arrival of smart-grid related devices is also modeled with a uniform-based process.

The parameters' settings are listed in table 1. The duration of the simulation corresponds to the distribution period of the most critical traffic (i.e. class 1). More details on the traffic patterns can be found in [13].

Table 1: Simulation parameters

Parameters	Values
Simulation Time	10s
Number of eNBs	1
Total number of preambles	54
Cell bandwidth	5MHz
Maximum number of preamble retransmissions	10
ac-BarringTime	4s
Total number of MTC devices	10000
Number of devices in class 1	1000
Number of devices in class 2	3000
Number of devices in class 3	6000
Beta distribution time	10s
Beta function parameters	$\alpha = 3; \beta = 4$

4.2 Simulations results

To validate the proposed model defined in (2), we will present, in this subsection, the analytical values of success and collision probabilities against the ones obtained using simulation. Then, we will give the number of successful ACB tests compared with the targeted value (i.e. optimal value as found in section 3.2) to demonstrate the effectiveness of our proposal. Finally, we show the efficiency of the proposed solution in prioritizing the different M2M applications and by comparing the estimated and the measured values of $x_{1,i}$, for all $i \in \mathbb{C}$.

Figures 5 and 6 show the simulation results obtained respectively for the success probability and the collision probability against those obtained theoretically. It can be seen that the simulation results match very well the theoretical values, which allow validating the proposal model.

Figure 7 illustrates the instantaneous and the average (i.e. EWMA) number of successful ACB tests. We can easily see

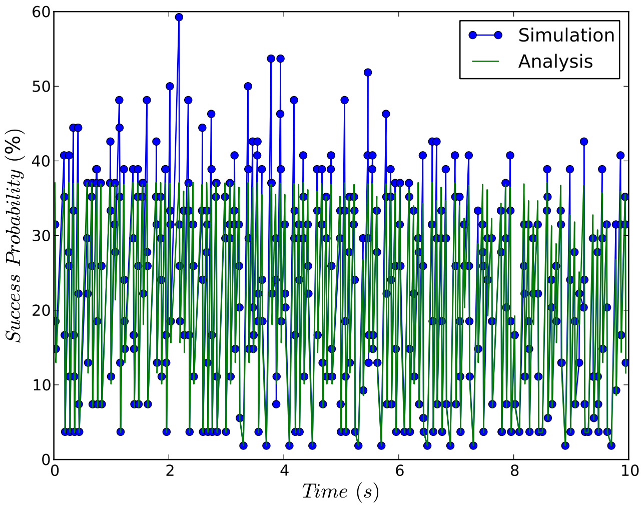

Figure 5: Success Probability

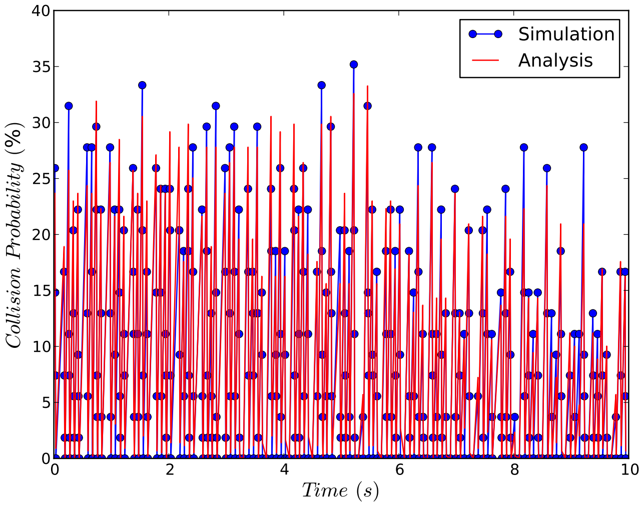

Figure 6: Collision Probability

that even if the instantaneous values oscillate between 30 and 70, the average values remain very close to the targeted value (i.e. 51), which is the objective of our mechanism. This, clearly, demonstrates the effectiveness of the PID controller as it helps in regulating the ACB factor dynamically according to the congestion level and the M2M application classes.

Note that values smaller than the target may lead to resources' underutilization, while values bigger than the target may lead to excessive collisions, which also lead to resources' underutilization.

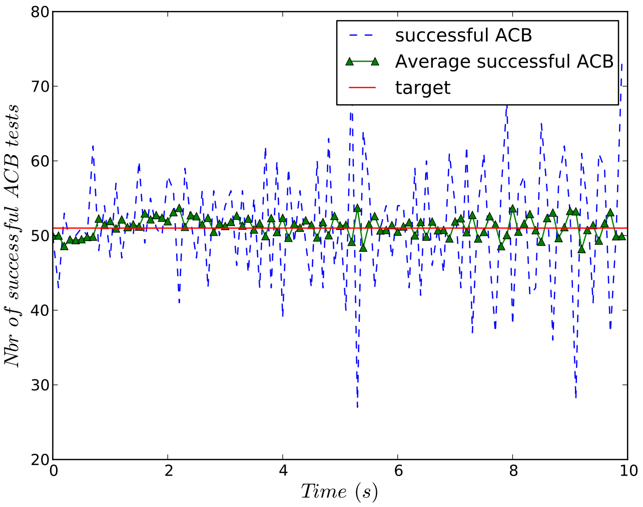

Figure 7: Total successful ACB tests vs Setpoint

To see the efficiency of the proposed algorithm in prioritizing different classes of M2M applications, let see the results obtained in Fig. 8. Figure 8 depicts the cumulative number of successful ACB tests for each class of M2M applications in figures (B) compared with the cumulative number of arrivals for each class in figures (A). It can be easily seen that the number of arrivals for class 1 is nearly equal to the number of successful ACB tests, which is in a complete conformance with the absolute constraint requirements for this category of traffic.

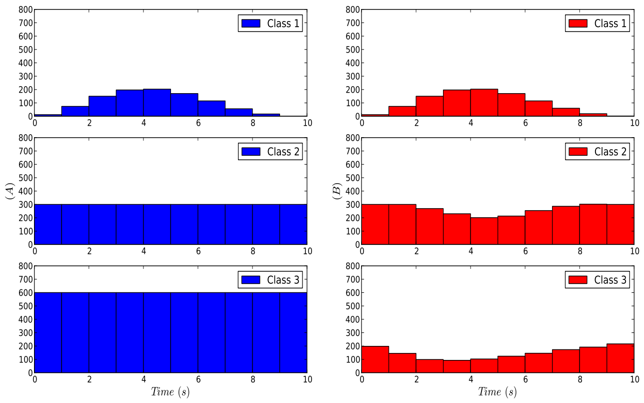

Figure 8: (A) Cumulative number of arrivals per second (B) Cumulative number of successful ACB tests per second

Another important observation, in Fig. 8, is that when the congestion is at its maximum (between seconds 2 and 8) the traffic of class 2 suffers from some degradation as some of the traffic is blocked. However, the blocking probability of this traffic class, which has a weight bigger than class 3, is low. This has direct consequences on the large number of successful ACB tests for this class. On the other hand, the less prioritized class (i.e. class 3) suffer from the beginning from blocked traffic. The phenomenon is exacerbated when the congestion is at its peak, as it can be seen in Fig. 8. In fact, given that devices of class 3 have the lowest priority, they will be blocked in case of congestion, i.e. when arrivals of class 1 and class 2 are more important. Consequently, it results in reducing collision probability as it is observed in figure 6.

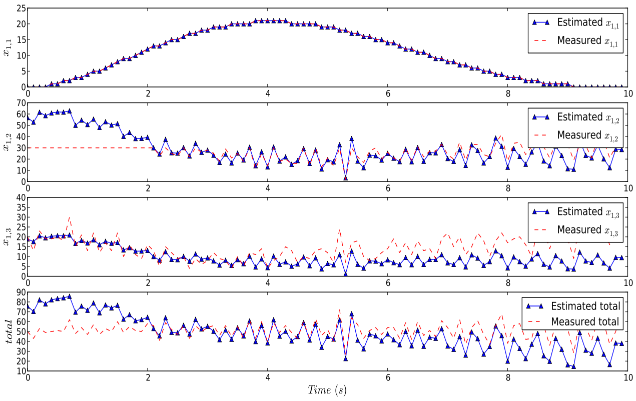

Figure 9: Estimated vs measured values of $x_{1,i}$, for all $i \in \mathbb{C}$

Another way to test the performances of the proposed algorithm can be achieved by comparing the estimated values of $x_{1,i}$, for all $i \in \mathbb{C}$ against the calculated ones. As illustrated in Fig. 9, it can be easily observed that estimated values for $x_{1,1}$ match the measured ones, which prove the efficiency of the estimation regarding the strict requirements of traffic of class 1. We can also observe a small shift between the estimated and the measured values for class 2. Such shift becomes larger for class 3, because of the number of ACB retries after the blocking duration (i.e. *Tbarring*) in case of congestion.

Another important observation, in Fig. 9, is that when the arrivals of class 1 are too small (between seconds 0 and 2), the estimated values of $x_{1,2}$ are bigger than the measured ones as the access is granted to all the devices of the second class. The constant values of measured $x_{1,2}$ (between seconds 0 and 2) are due to uniform traffic type of class 2.

5. CONCLUSIONS AND FUTURE WORK

In this paper, the RAN overload issue caused by MTC in LTE-Advanced networks has been addressed. We have proposed a novel mechanism as a way to alleviate RAN congestion by efficiently managing the M2M devices' random accesses.

The proposed access control mechanism constitutes a novel implementation of the ACB mechanism. To that purpose, we presented a fluid model with multiple M2M traffic classes in order to satisfy varied QoS requirements. This scheme utilizes multiple ACB factors each one representing the probability to compete for the access based on M2M class priority.

Using Monte-Carlo simulations, we find out the optimal number of the MTC devices that should compete for the random access to maximize the number of devices succeeding in the ACB procedure. After that, in order to regulate adaptively the ACB factor guaranteeing a total number of devices around the targeted value, we used a discrete PID controller. Then, we applied a scheduling algorithm as a way to schedule different M2M traffic classes.

In order to evaluate the performances of the proposed model, we have conducted simulation tests under ns3. Simulation results show that the proposed mechanism can accurately predict congestion situations by reducing the collision probability. Therefore, even if the arrival rates of the different supported classes become more important, the total successful ACB tests remains very close to the optimal value. The results, also, demonstrated the effectiveness of the proposed scheme as it allow respecting the strict QoS requirements of the most prioritized classes by blocking the MTC devices belonging to the less prioritized classes. This clearly allows reducing the congestion by insuring an increased RA success probability.

As future work, we plan to apply optimal non affine nonlinear control theory foundation to derive, directly, the optimal blocking probability for each class.

6. REFERENCES

[1] 3GPP TR 23.888 V1.6.0 (2011-11), Third generation partnership project; Technical specification group services and system aspects; "System improvements for machine-type communications," (Release 11).

[2] Geng Wu, Shilpa Talwar, Kerstin Johnsson, Nageen Himayat, and Kevin D. Johnson, "M2M: From Mobile to Embedded Internet," IEEE Communications Magazine, April 2011.

[3] 3GPP TS 36.321 V10.2.0 (2011-06): "Medium Access Control (MAC) protocol specification," Jun. 2011.

[4] A. Ksentini, Y. Hadjadj-Aoul, T. Taleb, "Cellular-based machine-to-machine: overload control," Network, IEEE , vol.26, no.6, pp.54,60, November 2012.

[5] 3GPP TS 36.331 V10.2.0: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC);" Protocol specification (Release 10). Jun. 2011.

[6] Lien SY, Liau TH, Kao CY, Chen KC, "Cooperative Access Class Barring for Machine-to-Machine Communications," IEEE Transactions on Wireless Communications, vol. 11, no. 1, pp. 27-32, January 2012.

[7] Yi-Huai Hsu, Kuochen Wang, Yu-Chee Tseng, "Enhanced cooperative access class barring and traffic adaptive radio resource management for M2M communications over LTE-A," published in Signal and Information Processing Association Annual Summit and Conference (APSIPA), Asia-Pacific, Oct. 29 2013-Nov. 1 2013.

[8] Duan S, Shah-Mansouri V, Wong VWS, "Dynamic Access Class Barring for M2M Communications in LTE Networks," submitted to IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, Dec. 2013.

[9] Bouzouita M, Hadjadj-Aoul Y, Zangar N, Tabbane S and Viho C, "A random access model for M2M

communications in LTE-advanced mobile networks," Modeling and Simulation of Computer Networks and Systems, Elsevier, 2015.

[10] Astrom K.J., Hagglund T., "Advanced PID Control," ISA-The Instrumentation, Systems, and Automation Society, ISBN 1556179421, 2006.

[11] Hang CC, Astrom KJ, Ho WK., "Refinements of the Ziegler Nichols tuning formula," IEEE Proc-D 1991;138(2).

[12] NS-3 Consortium. ns-3. Official website. Available at:"http://www.nsnam.org/"; 2015.

[13] 3GPP TR 37.868 V11.0.0 (2011-09) Third generation partnership project (3GPP). Technical specification group radio access network; "Study on RAN Improvements for Machine-type Communications," (Release 11).

[14] Petiz I, Salvador P, Nogueira A, "Characterization and modeling of M2M video surveillance traffic," The Fourth International Conference on Advances in Future Internet (AFIN), 2012.