Dynamic adaptive access barring scheme for heavily congested M2M networks

Meriam Bouzouita
University of Rennes 1, France
Mediatron, SUP'COM, Tunisia
mariem.bouzouita@supcom.tn

Yassine Hadjadj-Aoul
University of Rennes 1, France
yhadjadj@irisa.fr

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

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

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

ABSTRACT

The next mobile generation is planned to create a networked society, in which Machine-to-machine (M2M) communications are expected to play a key role in a plethora of domains and applications. The massive deployment of such devices may overwhelm the cellular network by imposing strong constraints on the Radio Access Network (RAN) because of the radio resources' scarcity. As the base station cannot accurately get the exact number of M2M arrivals, it cannot really predict the overload status, which limits any control action. Consequently, a better estimation of this number would efficiently help to overcome the risk of congestion. In this paper, we proposed a novel fluid model for M2M communications, which allows gaining an enhanced understanding of the dynamics of such systems. The provided analysis of the model was used to devise a new method to estimate accurately the number of M2M devices willing to connect as well as the number of random access attempts.We proposed, then, a novel implementation of the ACB process, which dynamically computes the ACB factor according to the network's overload conditions while includes a corrective action adapting the controller action based on the mismatch existing between the computed and the targeted mean load. Compared to existing approaches, the simulation results show that the proposed algorithms allow improving considerably the estimation of the number of M2M devices' arrivals, while outperforming existing techniques in terms of the number of abandons, the success and the collision probabilities, and the average access delay.

Keywords: M2M, Random access, fluid model, Congestion avoidance.

1. INTRODUCTION

The next mobile generation, is planned to create a networked and smart society in which the Internet of Things (IoT) will certainly play an important role [1]. Furthermore, the success of the forthcoming 5G standard is closely related to the efficient support of such devices, and particularly machine-to-machine (M2M) communications, which represent a key component of the IoT paradigm[2][3].

M2M communications, which will probably reach 2 billion units by 2020 [4], are expected to generate an enormous amount of traffic from various types of applications and services, such as smart grid, smart metering, intelligent transportation system, environmental monitoring, public safety, eHealth, etc [5]. This induces a diversity in M2M applications' requirements and traffic patterns, which make the handling of such devices even more complex. In fact, depending on their deployments, some applications are delay sensitive (e.g. eHealth, emergency situations) whilst others have strict requirements for low power consumption.

Owing to the massive deployment of M2M communications, in the near future, M2M devices may overwhelm the cellular network by causing congestion at both the Radio Access Network (RAN) and the Core Network (CN). Nonetheless, the RAN constitutes the most claiming part because of the radio resources' scarcity. In this way, new network access approaches are required to anticipate the congestion and the system overload by managing more efficiently the simultaneous Random Access (RA) of M2M devices, and hence tackling their explosive growth. This constitutes our main focus in this paper.

The congestion control problem due to competing M2M devices was considered very early as one of the priorities of the 3GPP. Indeed, the 3GPP standardizing body proposed many solutions to tackle such problem, including the Access Class Barring (ACB) concept [6], which is considered as one of the most efficient ways to tackle these types of congestion [7]. These works have been precursors of many other highly effective solutions, which succeeded in avoiding the congestion of the access network [8], [9]. However, these proposals present many limits when dealing with baseline congestion. Indeed, in such a condition, these techniques fail to avoid a synchronized access of M2M devices, which may results in some cases to a congestion collapse. Substantively, the base station cannot accurately estimate the number of M2M devices willing to connect and, hence, cannot predict the overload status. Consequently, a better estimation of this number would efficiently help to treat the congestion trouble.

The main issue addressed in this article is related to improving channel access for M2M communications in LTE-A

MSWiM'16 November 13-17, 2016, Malta.

networks and beyond. To face the identified challenges, we proposed a mechanism to better exploit channel utilization, a novel access protocol that adapts dynamically the access attempts of M2M devices according to the network congestion's level. Hence, we propose a new approach to estimate the RA attempts under the developed dynamic access algorithm, based on M2M new arrivals and backlogged equipment's estimation. A corrective action to the Proportional Integral Derivative (PID)[10][11] controller was added to better mismatch the estimated and the targeted payload.

The remainder of this paper is organized as follows. Section 2 gives an overview of the congestion issue in M2M networks followed by a brief description of the ACB mechanism. Section 3 is dedicated to describe a fluid model for M2M networks and to study the steady-state performance of such dynamic system. Section 4 portrays our new adaptive access protocol in addition to the proposed estimation algorithm. Section 5 is dedicated to the simulation setup and the analysis of our proposition. Finally, conclusions are presented in Section 6 with a summary recapping the main advantages and achievements of the proposed access protocol.

2. OVERVIEW OF THE CONGESTION PROBLEM IN M2M NETWORKS

Alleviating congestion situations is a major concern in view of the expected massive deployment of M2M devices in the future 5G standard. Indeed, the 3GPP stepped up its efforts to introduce several overload resolution mechanisms targeting different types of congestion in order to tackle the explosive growth of simultaneous M2M connections [12]. Hence, several PRACH overload resolution methods were defined to improve the support of such devices in LTE-Advanced networks [7].

To meet the performance requirements under excessive loads while avoiding useless connections' attempts of M2M devices, the ACB technique was adopted to deal with the associated challenges.

Using the ACB mechanism [6], the M2M devices start the RA process with a probability p, called the ACB factor, which should reflect the congestion situation. The devices that failed to access are backlogged during a random time, called the *Tbarring*, before they can retry again the ACB check. The *Tbarring* is given by the following equation:

$$Tbarring = (0.7 + 0.6 \times q) \times ac\text{-}BarringTime$$

where q represents a random number, in the interval $[0, 1]$, generated by the device after a failed ACB check, and the *ac-BarringTime* represents a fixed duration broadcasted by the eNodeB.

Each device having succeeded in the ACB process may attempt the RA by selecting randomly one preamble from the total number of available preambles N, which is a constant belonging to the interval $[4, 64]$ [13]. If this preamble is not chosen by any other equipment, the eNodeB indicates a successful preamble transmission, and consequently, the device passes correctly the first RA step. Otherwise, the RA attempt fails and the M2M equipment goes to a waiting state during a backoff time [14]. When the latter expires, the device reattempts the RA.

Note that, when reaching the maximum number of attempts R_{max}, the device abandons the transmission [14]. It may restart the access later from the beginning.

From a large set of research's interests, a particular focus was given to the dynamic determination of the ACB factor according to the collision level [6]. However, most of the proposed schemes turn out to be inefficient in case of heavily congested M2M networks. In fact, in such a situation, the eNodeB cannot determine exactly the number of devices attempting the ACB check nor the number of devices attempting the random access. This may lead to an inadequate action of the proposed schemes (i.e. too small or too high ACB barring factor), which may end up with a congestion collapse or resources' under-utilization. In this way, many methods were proposed in the literature to estimate accurately the number of devices attempting the access [15][16][17][18]. The main idea behind these approaches is to estimate the number of contending devices based on the average status of the preambles. The resulted estimation is then used to adjust properly the ACB factor or even to deactivate it as proposed in [16].

In the present paper, we propose a new comprehensive approach to adapt dynamically the ACB factor using an adaptive version of the PID [10][11] controller and a more accurate estimation of the number of random access attempts thanks to a continuous observation of the system's input (i.e. ACB factor) and the estimated outputs.

In the following, we will propose a model of the access scheme described previously.

3. A MODEL FOR M2M NETWORKS

3.1 A simple fluid model

Having described briefly how M2M devices access the mobile networks, we now direct our focus on modeling the whole system using a simple fluid model, described above.

Note that, for the sake of simplification, the model represented in Fig. 1 does not consider the case where the M2M devices reach the maximal number of attempts. This phenomenon should be avoided in a properly dimensioned or controlled system. Indeed, an efficient controller should minimize the number of re-attempts, whatever the number of terminals, to maximize the resources utilization. This can be clearly observed that the proposed approaches presents only few abandons, in the performance evaluation section[1].

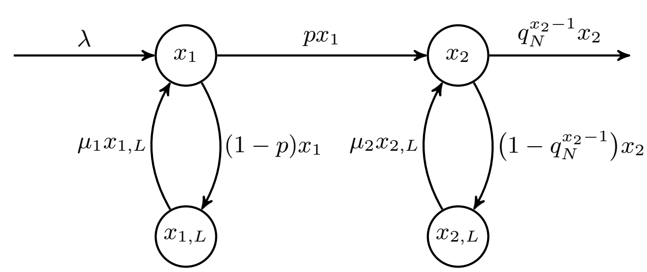


Figure 1: System model

The following quantities are used to capture the M2M devices behavior:

[1]This will be a part of a future work to consider the dynamics related to M2M devices abandoning the system.

$x_1(t)$ the number of backlogged M2M devices at time t;

$x_{1,L}(t)$ the number of blocked M2M devices after failed ACB check at time t;

$x_2(t)$ the number of M2M devices passing the ACB check and waiting to attempt the RA at time t;

$x_{2,L}(t)$ the number of blocked M2M devices after failed RA attempts at time t;

λ the arrival rate of M2M devices;

μ_1 the rate of ACB re-attempts;

μ_2 the rate of RA re-attempts;

p the ACB factor.

When attempting the random access, the M2M devices contend for the N available preambles. In each RACH opportunity, these preambles are split into successful (i.e. chosen by only one device), collided (i.e. chosen by two or more devices) and idle (i.e. selected by none of the devices) preambles. In the following, we recall the average values of these quantities that we determined in [8]. These quantities will be used in the proposed model and algorithms.

Let's define $q_N = 1 - 1/N$. The average number of successful preambles N_S, during the RACH opportunities is given as follows:

$$N_S = q_N^{x_2 - 1} x_2 \tag{1}$$

The average number of idle preambles is given by the following equation:

$$N_I = N q_N^{x_2} \tag{2}$$

From (1) and (2), we obtain the expected number of failed preambles N_F:

$$N_F = N - (N_S + N_I)\,. \tag{3}$$

Since the number of units considered in these systems is relatively high, we will use a fluid model (continuous state variables), that has the benefit of simplicity. In the sequel we will describe the dynamics that define the evolution of the four state variables x_1, $x_{1,L}$, x_2 and $x_{2,L}$ (all functions of time) based on the model described in Fig. 1.

The system's dynamics is described by the following system of differential equations:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= \lambda - x_1 + \mu_1 x_{1,L},\\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= p x_1 + \mu_2 x_{2,L} - x_2,\\
\frac{\mathrm{d}x_{1,L}}{\mathrm{d}t} &= (1-p)x_1 - \mu_1 x_{1,L},\\
\frac{\mathrm{d}x_{2,L}}{\mathrm{d}t} &= \big(1 - q_N^{x_2 - 1}\big)x_2 - \mu_2 x_{2,L}.
\end{aligned}$$

The considered parameters are the following: N with $4 \leq N \leq 64$, $\lambda > 0$, p with $0 < p < 1$, $\mu_1 > 0$, $\mu_2 > 0$ along with the obvious constraint that $x_1(t)$, $x_{1,L}(t)$, $x_2(t)$ and $x_{2,L}(t)$ should be non-negative.

3.2 Steady-state performance analysis

To study the steady-state performance of our system, we replace the derivatives by 0. We obtain the equations for finding the stationary points of the dynamical system:

$$0 = \lambda - \bar{x}_1 + \mu_1 \bar{x}_{1,L}, \tag{4}$$

$$0 = p\bar{x}_1 + \mu_2 \bar{x}_{2,L} - \bar{x}_2, \tag{5}$$

$$0 = (1-p)\bar{x}_1 - \mu_1 \bar{x}_{1,L}, \tag{6}$$

$$0 = \big(1 - q_N^{\bar{x}_2 - 1}\big)\bar{x}_2 - \mu_2 \bar{x}_{2,L}. \tag{7}$$

From (4) and (6), we obtain

$$\bar{x}_1 = \frac{\lambda}{p}.$$

Replacing the value of $\bar{x}_1$ in (4), we have

$$\bar{x}_{1,L} = \frac{1-p}{p}\frac{\lambda}{\mu_1}.$$

From (5), $\mu_2 \bar{x}_{2,L} = \bar{x}_2 - p\bar{x}_1$, and from (7), $\mu_2 \bar{x}_{2,L} = \left(1 - q_N^{\bar{x}_2 - 1}\right)\bar{x}_2$. Eliminating $\mu_2 \bar{x}_{2,L}$ and using the obtained expression of $\bar{x}_1$, we have

$$\bar{x}_2 - p\frac{\lambda}{p} = \left(1 - q_N^{\bar{x}_2 - 1}\right)\bar{x}_2,$$

leading to the equation

$$\bar{x}_2 q_N^{\bar{x}_2 - 1} = \lambda. \tag{8}$$

The analysis of this nonlinear equation is a bit out of the focus of the paper, but it is straightforward. We provide a short summary here following a direct approach, avoiding the classical use of the Lambert W function defined on any complex number z by $W(z) = w \iff z = we^w$ [19].

Define the function f by means of $f(x) = xq^{x-1}$ for $x \geq 0$, where $0 < q < 1$. We have $f(0) = f(\infty) = 0$ and $f'(x) = q^{x-1}(1 + x\ln q)$, giving a maximum at $x = x^* = -1/\ln q$, whose value is $M = -1/(\mathrm{e}q\ln q)$. This already gives us the stability condition of the "right side" of the model (the "left side", variables x_1 and $x_{1,L}$, is always stable): $\lambda \leq M$, if we use a generic q in the model, or $\lambda \leq -1/(\mathrm{e}q_N \ln q_N)$ for our specific value $q_N = 1 - 1/N$. If $\lambda < M$ then we have two solutions to the equation, say r_a and r_b, with $r_a < x^* < r_b$, where r_a leads to stability and r_b to instability. Knowing that $0 < r_a < x^*$ allows to easily find r_a numerically, for instance using a Newton scheme.

4. DYNAMIC ACB FOR HEAVILY OVERLOADED NETWORKS

In this section, we proposed a comprehensive solution, which consists in the two following phases. The first phase allows estimating accurately the states' space variables (i.e. x_1 and x_2) based on the mathematical model developed in the previous section. These variables are used, then, in the second phase to calculate dynamically the ACB factor using an adapted PID controller for an optimized resources management.

In contrast with existing approaches, the enhancements we suggest, in this paper, allow a more accurate estimation of the number of M2M devices attempting the random access through an iterative convergence of the estimates to the real values for an improved resources' management.

4.1 An accurate estimation of the states' space variables

In realistic use cases, the eNodeB is unaware of the devices present in the states x_1 and x_2, as no connection is established with them yet. To remedy this situation, we propose, in the following, a methodology allowing to estimate the number of devices in these two states (see Alg. 1 for more details).

For the sake of simplicity we are referring here by N_S, N_I and N_F to the average values calculated during 1 second (i.e. 100 RACH opportunities). The values N_S^*, N_I^*, and N_F^* represent the values obtained for the optimal x_2 (i.e. x_2^*).

Algorithm 1 States' estimation

1: **global:** $\hat{x}_1, \hat{x}_2, \bar{x}_2, x_2^*, N_S, N_S^*, N_F, N_I, N_I^*, P_{\text{acb}}$
2:
3: **function** x2Estimate(n)
4: **require:** $\delta, \eta, \alpha_1 \in [0, 1], \alpha_2 > 1$
5: $x_{2,min}[n] \leftarrow N_S[n] + 2N_F[n]$
6: **if** $N_I[n] \neq 0$ **then**
7: $x_{2,\text{idle}}[n] \leftarrow \frac{\ln(\frac{N_I[n]}{N})}{\ln q_N}$
8: $\tilde{x}_2[n] \leftarrow max(x_{2,min}[n], x_{2,\text{idle}}[n])$
9: **else**
10: **if** $N_S[n] > 20$ **then**
11: $N_S[n] = 19$
12: $x_{2,\text{success}}[n] \leftarrow \frac{W(q_N \ln q_N N_S[n])}{\ln q_N}$
13: $\tilde{x}_2[n] \leftarrow max(x_{2,min}[n], x_{2,\text{success}}[n])$
14: $e[n] \leftarrow e[n-1] + \delta(x_2^* - \tilde{x}_2[n])$
15: $\bar{x}_2 \leftarrow (1-\eta)\bar{x}_2 + \eta\tilde{x}_2[n]$
16: **if** $N_S > N_S^*$ **then**
17: $N_S = N_S^*$
18: $correction \leftarrow -N\frac{N_S - N_S^*}{N_S^*}$
19: $N_{I,min} \leftarrow \alpha_1 N_I^*$
20: $N_{I,max} \leftarrow \alpha_2 N_I^*$
21: **if** $N_I \leq N_{I,min}$ **then**
22: **return** $\hat{x}_2[n] \leftarrow \bar{x}_2 + e[n] + correction$
23: **else**
24: **if** $N_I \geq N_{I,max}$ **then**
25: **return** $\hat{x}_2[n] \leftarrow \bar{x}_2 + e[n] - correction$
26: **else**
27: **return** $\hat{x}_2[n] \leftarrow \bar{x}_2 + e[n]$
28:
29: **function** x1Estimate(n)
30: **require:** $\epsilon = 10^{-6}$
31: **if** $P_{\text{acb}[n-1]} < \epsilon$ **then**
32: **return** $\hat{x}_1[n] = \hat{x}_1[n-1]$
33: $\hat{x}_1[n] \leftarrow \frac{\hat{x}_2[n] - (\hat{x}_2[n-1] - N_S[n])}{P_{\text{acb}}[n-1]}$
34: **return** $\hat{x}_1[n] > 0 ? \hat{x}_1[n] : \hat{x}_1[n-1]$

During each RACH opportunity n, the eNodeB calculates the number of successful preambles $N_S[n]$, the idle ones $N_I[n]$ and the collided ones $N_F[n]$. Given that the number of successful preambles is also equal to the expected number of devices accomplishing the RA process, and given that a failed preamble is chosen by at least two M2M devices, one can deduce the minimal x_2 value (i.e. $x_{2,min}$), as expressed in line 5. In presence of at least one idle preamble (i.e. $N_I \neq 0$), the x_2 estimate, denoted by $x_{2,\text{idle}}$, is obtained using the solution of equation (2). Otherwise, it is estimated (i.e. $x_{2,\text{success}}$) using the solution of equation (1) with the constraint that $N_S[n] <= 19$ in order to guarantee a real solution of the obtained Lambert W function [19], which is analyzed in the previous section (see 3.2). This will allow calculating the value of $\tilde{x}_2$ and its moving average $\bar{x}_2$. Therefore, the mismatch existing between $\tilde{x}_2$ and the optimal value x_2^* can be computed to correct the value of $\hat{x}_2$.

Note that a corrective action is added when the average number of idle preambles N_I during the last second is bigger or smaller than predefined threshold.

Once x_2 estimation is accomplished, the eNodeB can use $\hat{x}_2$ and N_S to estimate x_1 according to the code in line 33.

4.2 Adaptive ACB calculation

In this subsection, we will describe in details the proposed adaptive ACB algorithm, named DACB, which is performed at each RACH opportunity and illustrated in Alg. 2.

Algorithm 2 Dynamic ACB calculation (DACB)

global: $\hat{x}_1, \hat{x}_2, \bar{x}_2, x_2^{\text{ref}}, x_2^*$

function refUpdate(n)
require: $\theta, \beta \in [0, 1], \alpha > 1$
$\bar{x}_2 \leftarrow (1-\theta)\bar{x}_2 + \theta\hat{x}_2[n]$
if $\bar{x}_2 > \alpha x_2^*$ **then**
$x_2^{\text{ref}}[n] \leftarrow x_2^{\text{ref}}[n-1] - 1$
else
if $\bar{x}_2 < \beta x_2^*$ **then**
$x_2^{\text{ref}}[n] \leftarrow x_2^{\text{ref}}[n-1] + 1$
$x_2^{\text{ref}}[n] \leftarrow \min\left(\max\left(x_2^{\text{ref}}[n], 0\right), x_2^*\right)$

procedure DACB
$n \leftarrow 1$
$x_2^* \leftarrow N$
loop:
$\hat{x}_2[n] \leftarrow$ x2Estimate(n)
$\hat{x}_1[n] \leftarrow$ x1Estimate(n)
refUpdate(n)
if $(\bar{x}_2[n] \leq x_2^*)$ **and** $(\hat{x}_1[n] \neq \text{NaN})$ **then**
$P_{\text{acb}}[n] \leftarrow \frac{x_2^{\text{ref}}[n]}{\hat{x}_1[n]}$
else
$e[n] \leftarrow x_2^{\text{ref}}[n] - \hat{x}_2[n]$
$P_{\text{acb}}[n] \leftarrow k_{\text{p}}e[n] + k_{\text{i}}\sum_{k=0}^{n} e[k] + k_{\text{d}}(e[n] - e[n-1])$
$P_{\text{acb}}[n] \leftarrow \min\left(\max\left(P_{\text{acb}}[n], 0\right), 1\right)$
$n \leftarrow n + 1$
goto *loop*

In the first phase of the DACB algorithm, the eNodeB adjusts dynamically the set point x_2^{ref} according to the overload situation. The main idea here is to compute the moving average value of the estimate $\hat{x}_2$ and to check if the value is bigger than the optimal value x_2^* (i.e. too much M2M devices in the state x_2) or smaller than it (i.e. too few M2M devices in the state x_2). To provide more stability, an action is taken only when the average is bigger or small than predefined thresholds. Thus, when the average value is considered as too big with a risk of congestion, the controller action reinforced by reducing the targeted objective x_2^{ref}, which enables blocking more devices from attempting the random access in the subsequent step. Besides, when the average value is considered as too small with a risk of resources' under-utilization, the controller action is relaxed by increasing the targeted objective, which enables accepting more devices. Note that we consider only values within the interval $[0; x_2^*]$.

Once the dynamic targeted load determined, the eNodeB executes the second phase to generate the ACB factor at step n. If the average of x_2 is less than x_2^* at step n, P_{acb} in the next step, is computed using equation in line 21, based on the estimation of $\hat{x}_1$. Otherwise, the eNodeB applies a PID controller to make the total number of M2M devices x_2, contending for RA, converges to the optimal value x_2^{ref} determined at step n. As P_{acb} is a probability, we apply: $\min(\max(P_{acb}, 0), 1)$ in (line 25).

Both the dynamic adjustment of the set point and the ACB factor generation are repeated in the following step (i.e. next RACH opportunity).

5. PERFORMANCE EVALUATIONS

5.1 Simulations' parameters

In this section, we evaluate the proposed solutions and highlight their technical benefits. In order to evaluate the accuracy of the estimations' and the efficiency of the dynamic ACB calculation's algorithm, we built a discrete events' simulator in C, which was validated in a previous study [8]. The developed simulator models the whole system described in section 3. Besides, we added the possibility for an M2M device to abandon the connection after reaching a maximal number of attempts R_{max}.

We assume that there is one eNodeB and that M2M devices are activated according to a Poisson traffic model where inter-arrivals are exponentially distributed. We also adopt an RACH configuration where one RACH opportunity occurs every $10ms$ with 54 preambles at each opportunity. The simulations' parameters are summarized in Table 1. When an RACH trial is declared unsuccessful, MTC device can retry the RA after a backoff time chosen uniformly between 0 and the backoff parameter fixed in the following table [14].

Table 1: Basic simulation parameters

Parameter	**Value**
Simulation duration	30s
Number of preambles N	54
x_2^*	54
R_{max}	10
Backoff parameter	20ms
ac-BarringTime	4s

In the following, we performed a series of experiments. In the first experiment, we evaluated both the estimation accuracy of the number of contending devices and the performance of the proposed adaptive controller as a function of the time. In the second set of experiments, we analyzed the accuracy of the estimation and the efficiency of the proposals for various rates.

5.2 Experiment 1

5.2.1 *Controller efficiency*

In Figure. 2, we represent the instantaneous and the mean number of RA attempts, as calculated in the previous section. In the simulation, we used an arrival rate following a Poisson process with an average arrival corresponding to 28 M2M devices per RACH opportunity (i.e. $10ms$). It can be seen that even if the instantaneous values present large fluctuations between 0 and 210, the average values remain close to the targeted value $x_2^* = 54$. The obtained average along all the simulation is equal to $\bar{x}_2 = 57.17$. Let's recall that having a number of devices around the optimal value guarantee an optimized usage of resources.

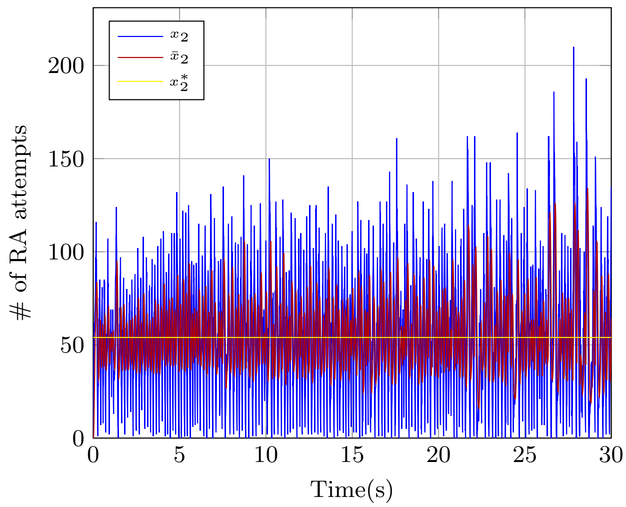


Figure 2: Number of RA attempts as a function of time

Figure 3 illustrates the evolution of the targeted number of M2M devices in state x_2 (i.e. x_2^{ref}) as a function of time. It can be easily seen that the set-point varies dynamically depending upon the overload situations, which demonstrates the robustness of DACB algorithm. Indeed, at the beginning of the simulation the number of devices in x_2 is moderated, which keep the targeted value high. However, when the accumulated number of devices starts to be very important, the set-points decreases. Thus, it corresponds to the peak number of devices in Fig. 2, the smallest value of the set-point, as it can be seen in Fig. 3.

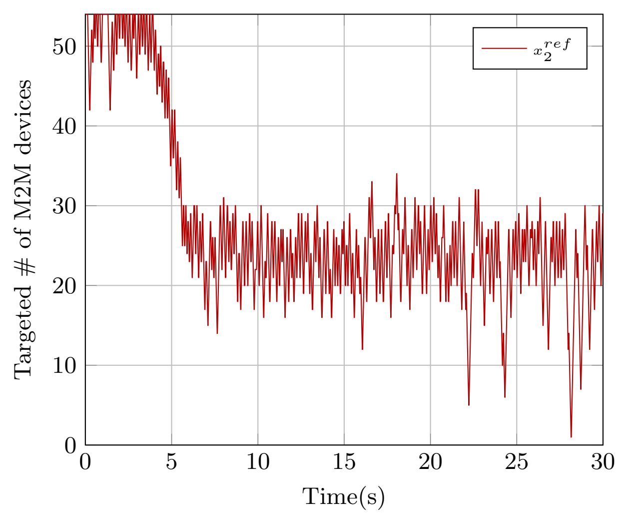


Figure 3: Targeted number of M2M devices as a function of time

Another way to test the efficiency of our proposed algorithm can be reached by computing the number of devices withdrawal (i.e. after reaching the maximum number of RA re-transmissions R_{max}). As illustrated in Figure 4, we can easily remark that the number of abandons is very small. It correspond, indeed, to an average number of 0.23 abandons per second, which was one of the objectives of our proposed solution. It can be noted that the number is at its maximum when the number of devices is very huge (see Fig. 2).

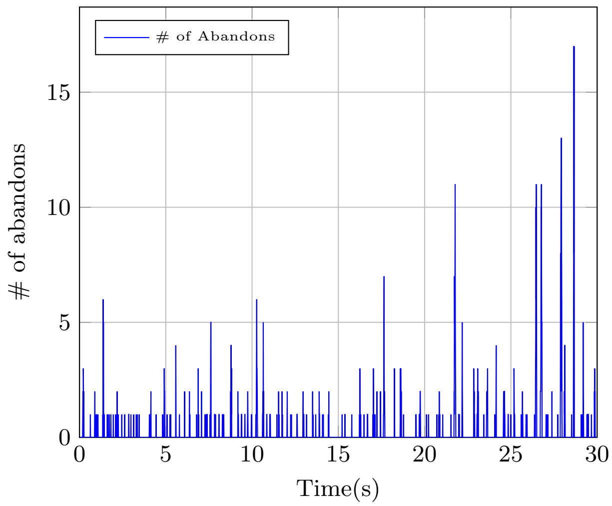


Figure 4: RA abandons

5.2.2 *Estimation accuracy*

Fig. 5 shows the gap existing between the real number x_1 of devices attempting the ACB and the estimated one $\hat{x}_1$. It can be seen that the estimation error is sometimes large even it presents an average value equal approximately to 23.72%, which can be considered as reasonable as no existing work, in our knowledge, proposes this estimation. Note that the estimation error is smaller than the estimation error for x_2 when predicted in [20].

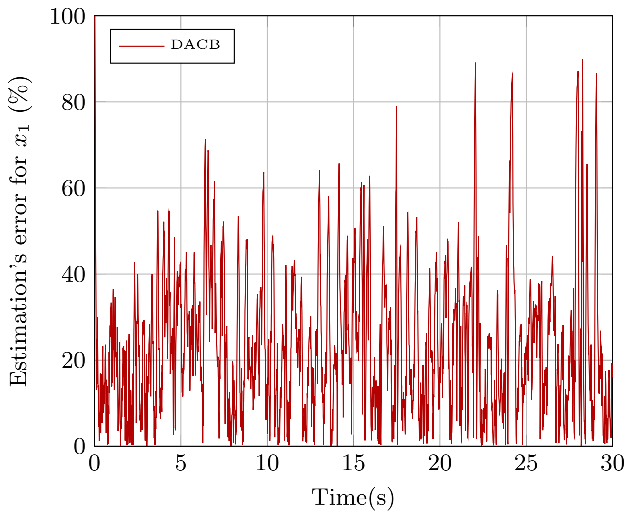
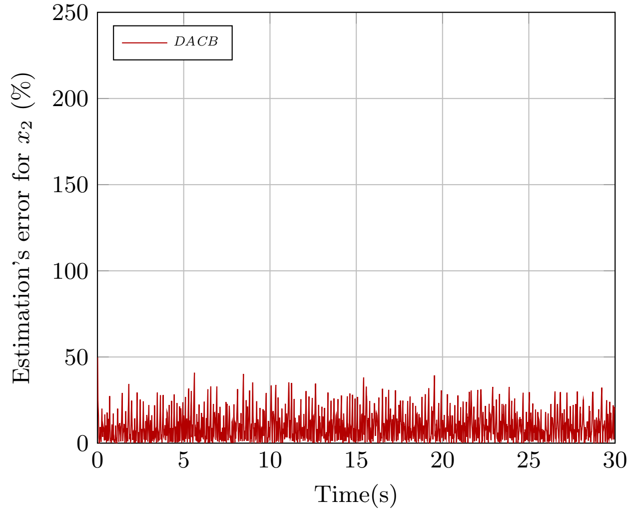


Figure 5: Estimation error of the number of ACB attempts as a function of time

Fig. 6 shows the estimation error between the number x_2 and the estimated one $\hat{x}_2$. We can easily remark from the figure that the behaviour of our estimation scheme clearly outperforms *MCSA-OE* scheme [20]. In fact, we obtained an average deviation of 9.95% against 30.85% in case of *MCSA-OE*, which presents more oscillations and many peaks exceeding 50%.

(a)



(b)

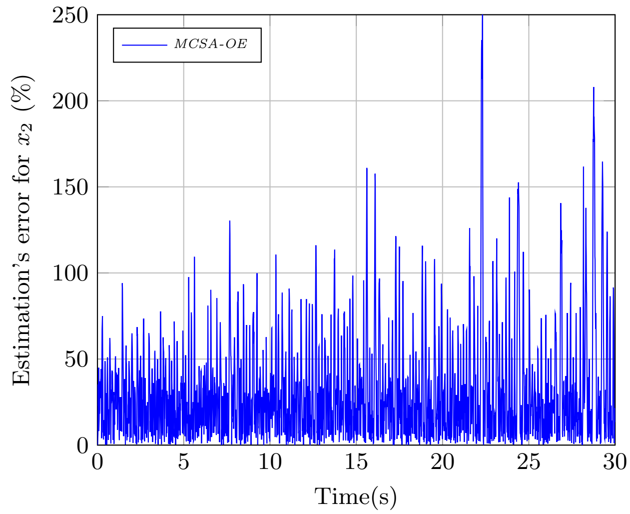


Figure 6: Estimation error of the number of RA attempts as a function of time

5.3 Experiment 2

5.3.1 *Controller efficiency for various loads*

To show the behaviour of our proposed DACB algorithm, we compare, in this sub-section, its performance with the PID controller for various network's loads [10]. Let's recall that the average load is represented here for a period of $10ms$. We consider the following performance's metrics: the average number of abandons, the average number of successful RA attempts and the average access delay. Here, we define the random access delay as the duration from the first RA attempt until a successful access. The obtained average values and the confidence intervals were computed for 50 experiments for each load value.

Fig. 7 depicts the average number of abandons for different network loads. In relaxed network conditions, the results obtained with the PID controller are comparable to our approach (i.e. *DACB*) and remain acceptable. Indeed, in such conditions there is no impact of the proposed adaptation mechanism, and the two approaches have the same behavior. When considering *DACB*, we note that this number remains very close to 0 even if the network's load increases, which proves the effectiveness and the stability of DACB. It can also be seen that the number of abandons increases rapidly when applying the PID controller.

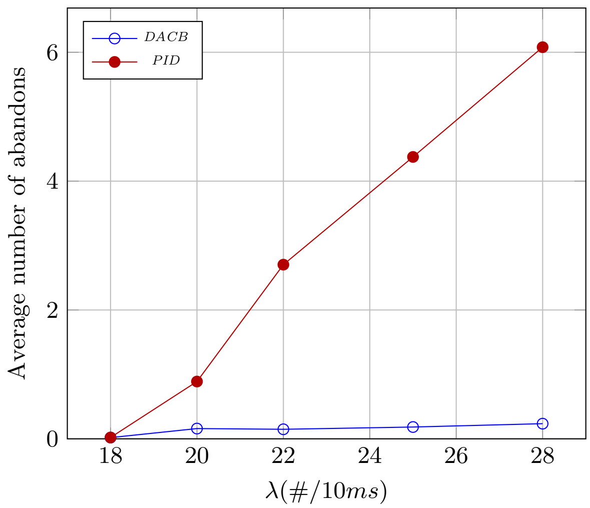


Figure 7: Evolution of the number of abandons vs λ

Another way to test the performance improvements achieved by *DACB* can be reached by comparing the numbers of successful RA attempts in function of network loads. The obtained results are depicted in Fig. 8. We first observe a very small variation, with *DACB*, of these numbers even if the number of M2M devices increases (≥ 16). Whereas, in case of the PID controller, the number of successful RA attempts decreases gradually and becomes intolerable (around 5 devices per RACH opportunity), when the network is undergoing a heavy congestion. Nevertheless, this number remains acceptable when the network is in relaxed conditions.

Another important performance parameter is the average random access delay illustrated in Fig. 9. If DACB method is applied, we observe that the average delays for different loads are not much different and don't exceed $45ms$. Whereas, with the PID controller, we can easily observe that the delays reach $80ms$ when the network is in heavily congested situations. This is a direct consequence of the important number of connections' reattempts, which is also reflected by the number of abandons as it can be seen in Fig. 7.

5.3.2 *Estimation accuracy for various loads*

We compare, in this sub-section, the accuracy of the proposed scheme with the *MCSA-OE* algorithm [20] for various network's loads.

Fig. 10 depicts the evolution of the estimation error for various loads. It can be clearly seen that when the network's load increases, the average estimation error decreases and

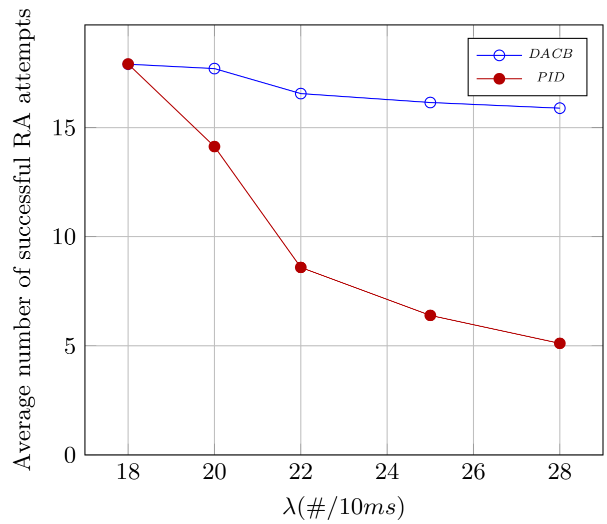


Figure 8: Evolution of the number of successful RA attempts vs λ

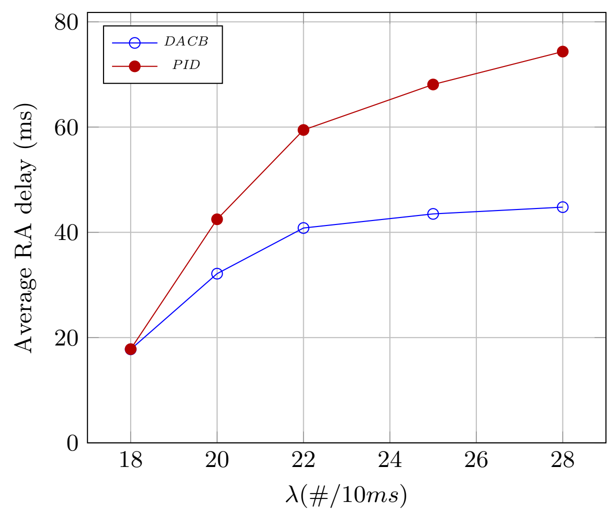


Figure 9: Evolution of RA delay vs λ

then we obtain a more accurate estimation. In fact, when the network is more congested, x_2 tends to reach stable values and then the estimation's error is reduced. Nevertheless, the obtained average deviation remains very small near to 0 when a *DACB* algorithm is considered. However, this fluctuation varies between 20% and 50% in case of *MCSA-OE* scheme.

Note that the error bars are obtained for 95% confidence intervals, which show the accuracy of the obtained values.

6. CONCLUSIONS

In this paper, we have addressed the issue of heavily congested M2M networks, where a risk of congestion collapse appears. This congestion is mainly related to the estimation's fluctuations of the number of M2M arrivals. Indeed, even if an efficient overload's resolution controller is applied, its performance jointly depends on its estimation's accuracy.

To treat more efficiently this trouble and improve the network's performances, we have proposed a new access

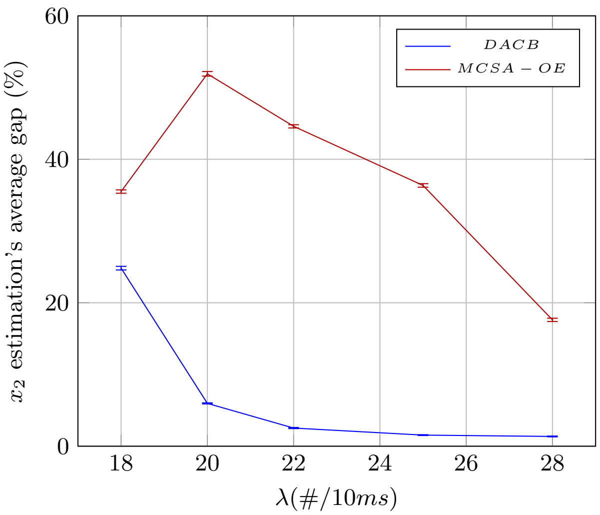


Figure 10: Evolution of the estimation error of x_2 vs λ

control strategy for M2M random accesses.

We first presented a simple fluid model for M2M devices' accesses. Then, based on this mathematical model, we designed a novel implementation of the ACB scheme, which combines three steps performed every RA opportunity: (i) an accurate estimation of the network status, (ii) a dynamic adjustment of the model's parameters depending on the RA congestion level (e.g. the number of RA attempts that maximize the success access probability) and (iii) finally a dynamic ACB probability's calculation according to the expected network's overload situations.

The simulation results demonstrated the efficiency of our proposition as it allows reducing the congestion probability while maximizing the success access probability compared to existing approaches. In fact, our proposition outperforms clearly both the classical ACB method and the PID controller. Additionally, the results showed a reduced random access delay and also a reduced number of RA preambles' re-transmissions which is one of the most important factors impacting the M2M energy consumption. Furthermore, results proved the efficiency of the proposed estimation method, as we obtained estimated values near to the actual ones.

As future works, we plan to direct our research to M2M energy aspects where it will be interesting to know the efficiency of our proposal regarding the improvement of M2M device's battery consumption.

7. REFERENCES

[1] M. R. Palattella et al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510-527, March 2016.

[2] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, M. Zorzi, *The challenges of M2M massive access in wireless cellular networks, Digital Communications and Networks*, Volume 1, Issue 1, Pages 1-19, 2015.

[3] R. Ratasuk, A. Prasad, L. Zexian, A. Ghosh, M. Uusitalo, *Recent advancements in M2M communications in 4G networks and evolution towards 5G*, Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on , vol., no., pp.52,57, 2015.

[4] Cisco VNI blog post, *2015 Cisco VNI Complete Forecast Update: Key Trends Include Mobility, M2M, and Multimedia Content.*

[5] Shafiq, M.Z., and al., *A First Look at Cellular Machine-to-Machine Traffic: Large Scale Measurement and Characterization*, ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 1, 2012.

[6] 3GPP TS 36.331 : *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)*; Protocol specification V.12.5.0 Release 10, April 2015.

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

[8] Bouzouita M, Hadjadj-Aoul Y, Zangar N, Rubino G and Tabbane S, *Multiple Access Class Barring factors Algorithm for M2M communications in LTE-Advanced Networks*, ACM/MSWIM, 2015.

[9] Arouk O., Ksentini A. and Taleb T., *Group Paging-based Energy Saving for Massive MTC Accesses in LTE and Beyond Networks*, IEEE Journal on Selected Areas in Communications, 2016.

[10] 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.

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

[12] 3GPP TR 37.868, *RAN Improvements for Machine-type Communications*, V.11.0.0, March 2013.

[13] 3GPP TS 36.331, *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)*, Protocol specification V.12.5.0 Release 10, April 2015.

[14] 3GPP TS 36.321, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification*, V.11.3.0 Release 11, July 2013.

[15] Hongliang He, Qinghe Du, Houbing Song, Wanyu Li, Yichen Wang, and Pinyi Ren, *Traffic-aware ACB scheme for massive access in machine-to-machine networks*, IEEE ICC, 2015.

[16] Y. Pang, G. Lin and H. Wei, *Context-aware Dynamic Resource Allocation for Cellular M2M Communications*, Internet of things journal, 2015.

[17] Hongliang He, Pinyi Ren, Qinghe Du, Li Sun, *Estimation Based Adaptive ACB Scheme for M2M Communications*, Wireless Algorithms, Systems, and Applications book, pp 165-174, 2015.

[18] German Corrales Madueno, Nuno K. Pratas, Cedomir Stefanovic, Petar Popovski, *Massive M2M Access with Reliability Guarantees in LTE Systems*, IEEE ICC, 2015.

[19] Corless, R., Gonnet, G., Hare, D., Jeffrey, D. and Knuth, D., *On the Lambert W function*, Advances in Computational Mathematics (Berlin, New York: Springer-Verlag) 5: 329–359, 1996.

[20] Arouk O. and Ksentini A., *Multi-Channel Slotted Aloha Optimization for Machine-Type-Communication*, ACM/MSWiM, 2014.