Dynamic Adaptive Access Barring Scheme For HeavilyCongested 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 massive deployment of Machine-to-machine (M2M) communications may overwhelm the cellular network by imposing strong constraints on the Radio Access Network (RAN). As the base station cannot accurately get the exact number of M2M arrivals, it cannot really predict the overload status. 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. 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. The simulation results show that the proposed algorithms allow improving considerably the estimation of the number of M2M devices' arrivals, while outperforming existing techniques.

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].

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 [4], which is considered as one of the most efficient ways to tackle these types of congestion [5]. These works have been precursors of many other highly effective solutions, which succeeded in avoiding the congestion of the access network [6], [7]. 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 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)[8][9] controller was added to better mismatch the estimated and the targeted payload.

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

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

© 2016 ACM. ISBN 978-1-4503-4502-6/16/11…$15.00

DOI: http://dx.doi.org/10.1145/2988287.2989174

2. A MODEL FOR M2M NETWORKS

2.1 A simple fluid model

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

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. In the performance evaluation section[1], it can be clearly observed that the proposed approach presents only a few abandons.

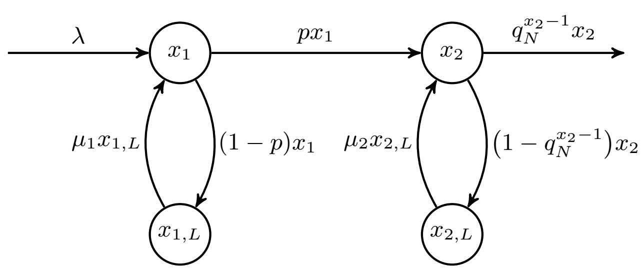

Figure 1: System model

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

- $x_1(t)$ number of backlogged devices at time t;
- $x_{1,L}(t)$ number of blocked devices at time t, after having failed an ACB check and waiting for re-attempting;
- $x_2(t)$ number of devices at time t having passed the ACB check and waiting to attempt the RA;
- $x_{2,L}(t)$ the number of blocked devices at time t after a failed RA attempt and waiting to try again;
- λ the arrival rate of devices;
- θ the abandon rate after reaching maximum number of RA attempts.
- 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). In the following, we recall the average values of these quantities that we determined in [6]. 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 by $N_S = q_N^{x_2-1}x_2$. The average number of idle preambles is $N_I = Nq_N^{x_2}$. From these expressions, we obtain the expected number of failed preambles N_F just by writing $N_F = N - (N_S + N_I)$.

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:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= \lambda - x_1 + \mu_1 x_{1,L},\\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= px_1 + \mu_2 x_{2,L} - x_2,\\
\frac{\mathrm{d}x_{1,L}}{\mathrm{d}t} &= (1-p)x_1 - \mu_1 x_{1,L},\\
\frac{\mathrm{d}x_{2,L}}{\mathrm{d}t} &= \left(1-q_N^{x_2-1}\right)x_2 - \mu_2 x_{2,L}.
\end{aligned}$$

The considered parameters are the following: N with $4 \le N \le 64$, $\lambda > 0$, 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.

2.2 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:

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

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

$$0 = (1-p)\bar{x}_1 - \mu_1 \bar{x}_{1,L}, \tag{3}$$

$$0 = \left(1-q_N^{\bar{x}_2-1}\right)\bar{x}_2 - \mu_2 \bar{x}_{2,L}. \tag{4}$$

From (1) and (3), we obtain

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

Replacing the value of $\bar{x}_1$ in (1), we have

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

From (2), $\mu_2\bar{x}_{2,L} = \bar{x}_2 - p\bar{x}_1$, and from (4), $\mu_2\bar{x}_{2,L} = \left(1-q_N^{\bar{x}_2-1}\right)\bar{x}_2$. 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

$$\bar{x}_2 q_N^{\bar{x}_2-1} = \lambda. \tag{5}$$

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 $W(z) = w \iff z = we^w$ [11].

Define the function f by means of $f(x) = xq^{x-1}$ for $x \ge 0$, 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): $\lambda \le M$, if we use a generic q in the model, or $\lambda \le -1/(\mathrm{e}q_N \ln q_N)$ for our

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

specific value $q_N = 1 - 1/N$. If $\lambda < M$ then we have two solutions r_a and r_b to the equation, say $0 < r_a < x^* < r_b$, 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.

3. 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.

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:     $\hat{x}_1[n] \leftarrow \dfrac{\hat{x}_2[n] - (\hat{x}_2[n-1] - N_S[n])}{P_{\text{acb}}[n-1]}$
34:     **return** $\hat{x}_1[n] > 0 ? \hat{x}_1[n] : \hat{x}_1[n-1]$

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.

3.1 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^*).

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 to equation $N_I = Nq_N^{x_2}$, seen in previous subsection. Otherwise, it is estimated (i.e. $x_{2,\text{success}}$) using expression $N_S = q_N^{x_2 - 1}$ with the constraint that $N_S[n] <= 19$, in order to guarantee a real solution of the obtained Lambert W function [11], which is analyzed in previous Subsection 2.2. 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.

3.2 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.

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}

Algorithm 2 Dynamic ACB calculation (DACB)

1: **global:** $\hat{x}_1, \hat{x}_2, \bar{x}_2, x_2^{\mathrm{ref}}, x_2^*$
2:
3: **function** REFUPDATE(n)
4: **require:** $\theta, \beta \in [0, 1], \alpha > 1$
5: $\bar{x}_2 \leftarrow (1-\theta)\bar{x}_2 + \theta\hat{x}_2[n]$
6: **if** $\bar{x}_2 > \alpha x_2^*$ **then**
7: $x_2^{\mathrm{ref}}[n] \leftarrow x_2^{\mathrm{ref}}[n-1] - 1$
8: **else**
9: **if** $\bar{x}_2 < \beta x_2^*$ **then**
10: $x_2^{\mathrm{ref}}[n] \leftarrow x_2^{\mathrm{ref}}[n-1] + 1$
11: $x_2^{\mathrm{ref}}[n] \leftarrow \min\left(\max\left(x_2^{\mathrm{ref}}[n], 0\right), x_2^*\right)$
12:
13: **procedure** DACB
14: $n \leftarrow 1$
15: $x_2^* \leftarrow N$
16: *loop*:
17: $\hat{x}_2[n] \leftarrow$ X2ESTIMATE(n)
18: $\hat{x}_1[n] \leftarrow$ X1ESTIMATE(n)
19: REFUPDATE(n)
20: **if** $(\bar{x}_2[n] \leq x_2^*)$ **and** $(\hat{x}_1[n] \neq \mathrm{NaN})$ **then**
21: $P_{\mathrm{acb}}[n] \leftarrow \frac{x_2^{\mathrm{ref}}[n]}{\hat{x}_1[n]}$
22: **else**
23: $e[n] \leftarrow x_2^{\mathrm{ref}}[n] - \hat{x}_2[n]$
24: $P_{\mathrm{acb}}[n] \leftarrow k_{\mathrm{p}} e[n] + k_{\mathrm{i}} \sum_{k=0}^{n} e[k] + k_{\mathrm{d}}(e[n] - e[n-1])$
25: $P_{\mathrm{acb}}[n] \leftarrow \min\left(\max\left(P_{\mathrm{acb}}[n], 0\right), 1\right)$
26: $n \leftarrow n + 1$
27: **goto** *loop*

determined at step n. As P_{acb} is a probability, we apply: $\min\left(\max\left(P_{acb}, 0\right), 1\right)$ 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).

4. PERFORMANCE EVALUATIONS

4.1 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 [6]. The developed simulator models the whole system described in section 2. Besides, we added the possibility for an M2M device to abandon the connection after reaching a maximal number of attempts $R_{max} = 10$.

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 $N = x_2^* = 54$ preambles at each opportunity. The simulation duration, the backoff parameter, the ac-BarringTime are respectively equal to 30s, 20ms and 4s [10].

4.2 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 [8]. 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. 2 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.

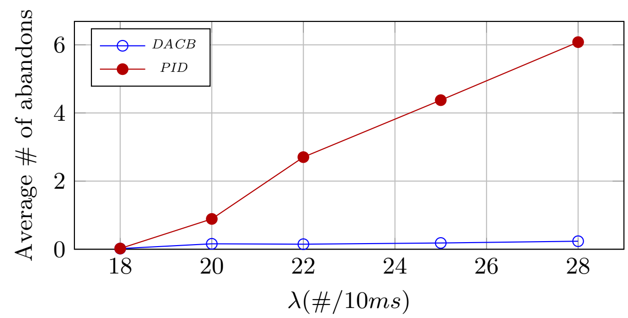

Figure 2: Evolution of the number of abandons versus λ

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. 3. 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.

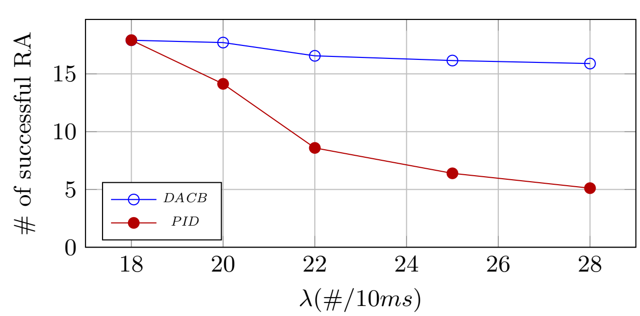

Figure 3: Evolution of the average number of successful RA attempts versus λ

Another important performance parameter is the average random access delay illustrated in Fig. 4. 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. 2.

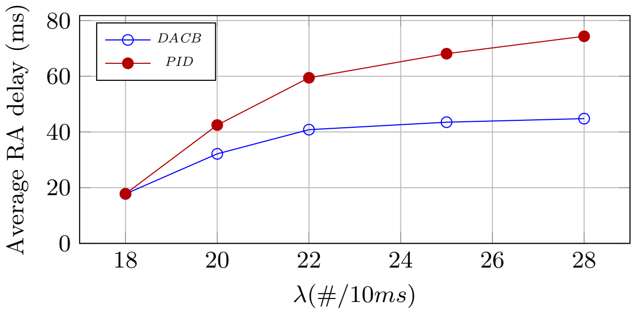

Figure 4: Evolution of RA delay versus λ

4.3 Estimation accuracy for various loads

We compare, in this sub-section, the accuracy of the proposed scheme with the *MCSA-OE* algorithm [12] for various network's loads.

Fig. 5 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 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.

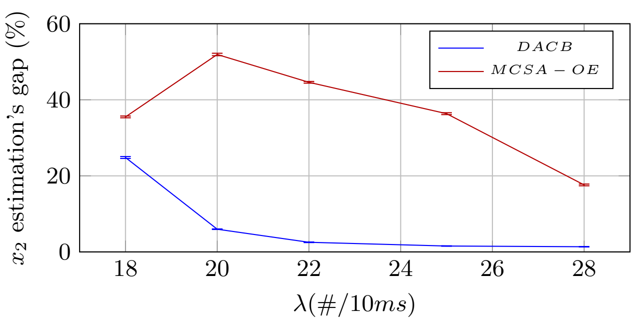

Figure 5: Evolution of the average estimation error of x_2 versus λ

5. CONCLUSIONS

In this paper, we have addressed the issue of heavily congested M2M networks, where a risk of congestion collapse appears. To treat more efficiently this trouble and improve the network's performances, we have proposed a new access 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 showed a reduced random access delay and also a reduced number of RA preambles' retransmissions 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.

6. REFERENCES

[1] M. R. Palattella et al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," in IEEE JSAC, 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 Com. and Networks*, Vol. 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*, ICIN 2015 , vol., no., pp.52,57, 2015.

[4] 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.

[5] 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.

[6] 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.

[7] Arouk O., Ksentini A. and Taleb T., *Group Paging-based Energy Saving for Massive MTC Accesses in LTE and Beyond Networks*, IEEE JSAC, 2016.

[8] 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.

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

[10] 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.

[11] 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.

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