Adaptive access protocol for heavily congested M2M networks

Meriam Bouzouita*, Yassine Hadjadj-Aoul†, Nawel Zangar‡, and Sami Tabbane§
*†University of Rennes 1, France
*‡§Mediatron, SUP'COM, Tunisia
Email: *mariem.bouzouita@supcom.tn, †yhadjadj@irisa.fr, ‡nawel.zangar@insat.rnu.tn,
§Sami.Tabbane@insat.rnu.tn

***Abstract*—Machine-to-Machine (M2M) communications are expected to be one of the major drivers of future cellular networks, due to a plethora of services provided to operators and consumers. This leads to an explosively growth of simultaneous M2M arrivals, and then a bursty random access attempts that causes a severe random access congestion. One of the key solutions to deal with such congestion is the Access Class Barring (ACB) scheme. However, this scheme presents many limits when dealing with baseline congestions. Indeed, in such conditions, it induces a synchronization of the terminals willing to connect, as we demonstrated in the paper, which results in poor performances. Thus, we proposed, in this paper, a new method to estimate accurately the number of random access attempts in a way to measure efficiently the congestion level. Then, we proposed a novel implementation of the ACB process that mainly consists of dynamically adapting the ACB factor according to the network's overload conditions. Simulation results show that the proposed algorithm outperforms the existing solutions by improving significantly the access's success probability while minimizing radio resources' underutilization. Keywords: MTC, M2M, Random Access, ACB, congestion, dynamic ACB factor.**

I. Introduction

Machine-to-Machine (M2M) communications, with their capability of providing various types of applications and services [3], are already considered to be a crucial technology for 4G LTE-Advanced systems. M2M communications are, also, expected to become a key enabler for the fifth generation (5G) of communications systems [4] [2]. Indeed, one of the most important challenges for Network Operators is to support more efficiently the Internet of things (IoT), which comprises M2M communications.

In contrast to Human-to-Human (H2H) scenarios, M2M scenarios require a set of challenges related to the support of a rapid increase number of M2M devices. In fact, M2M communications are expected to grow more and more during the forecast period between 2014 and 2019, reaching 10.5 billion connections by 2019 (up from 3.3 billion in 2014) [5]. Therefore, this numerous number of M2M devices may arrive simultaneously and ask for network connection establishment as a way to ensure data transmission between the eNodeB and each equipment [2].

To achieve that purpose, every device should first perform the RACH (Random access channel) procedure [6] [7]. It initially selects one of the available preambles' sequences. If the preamble is chosen by only that single device, the device can carry on with the RA (Random access) procedure. Otherwise, i.e. if two or more devices choose the same preamble sequence; a preamble collision is detected. Consequently, when the number of M2M devices becomes higher, this has a tremendous potential to generate a heavier random access traffic and then results in a high preambles' congestion.

To address this congestion issue, we propose, in this paper, a novel access protocol that adapts dynamically the access of M2M devices according to network's congestion's level. Then, we propose a new approach to estimate the RA attempts under the developed dynamic access algorithm.

The remainder of this article is organized as follows. In section 2, we describe the general architecture of the ACB mechanism followed by the RA process as introduced by the 3GPP. Section 3 is dedicated to expose the limitations related to our last contribution. Section 4 portrays our new adaptive access proposed protocol and the estimation algorithm. Section 5 is dedicated to the simulation setup and the discussion of the obtained results. Finally, the paper concludes in Section 6 with a summary recapping the main advantages and achievements of the proposed access protocol.

II. Related works

To alleviate the congestion problem and consequently, make the next cellular networks more suitable to ensure the efficient support of M2M services, the 3GPP standard introduces several overload resolution mechanisms [8] [9]. In literature, many resolution schemes were proposed based on the Access class barring (ACB) [10] method, as it helps reducing radio resources wastage caused when rejecting accesses from devices. Therefore, it guarantees resources' saving by dissuading some M2M devices from starting the RA.

The ACB principle defines a probability p called *ACBfactor* and a time-duration called $Tbarring$. When a new M2M RA attempt comes, the device goes to a first state x_1 and then starts the ACB check. It generates a random number q between 0 and 1. If q is less than p, M2M device passes successfully the ACB check and can, then start the RA procedure. Otherwise, it goes to a waiting state $x_{1,L}$ where it will be blocked during a $Tbarring$ duration before retrying a new ACB check. $Tbarring$ is given by the following equation:

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

where q_b represents a random number in the interval $[0, 1]$ generated by the device after a failed ACB check, and *ac-BarringTime* represents a fixed duration broadcast by eNodeB. Formerly the $Tbarring$ was expired, the M2M equipment returns to x_1 state and retries the ACB check [10].

Once the ACB check is successfully finished, M2M device goes to another state x_2 and then attempts the RA. It chooses one preamble from the total pool sequences denoted N available during one RACH opportunity i. If this preamble is not chosen by any other equipment, eNodeB indicates a successful preamble transmission, and consequently, the device passes correctly the first RA step. If not, RA attempt fails and the M2M equipment goes to a second waiting state $x_{2,L}$ during a back off time. When the latter is ceased, the device returns to state x_2 and can reattempt the RA in the next RACH opportunity while the maximum number of preamble re-transmissions denoted R_{max} is not reached [6]. The whole system explained previously, is depicted in Fig. 1.

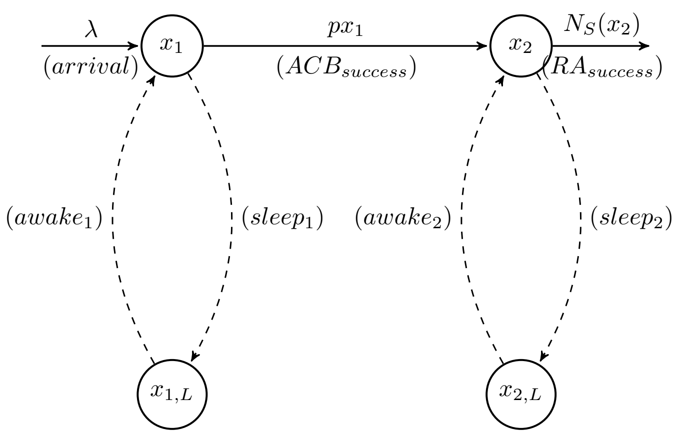

Fig. 1: System model

The predictable number of successful preambles N_S, during one RACH opportunity i is given as follows [11]:

$$N_S(i) = x_2\Big(1 - \frac{1}{N}\Big)^{x_2-1} \simeq x_2 e^{-\dfrac{x_2-1}{N}} \tag{1}$$

where x_2 depicted the total number of devices present at state X_2 in order to attempt RA during a RACH opportunity i. The expected number of failed preambles N_F can be derived given the following equation [11]:

$$N_F(i) = N - x_2 e^{-\dfrac{x_2-1}{N}} - Ne^{-\dfrac{x_2}{N}} \tag{2}$$

It can be easily checked that N_S is maximized when the number x_2 is equal to the total number of preambles N available during one RACH opportunity. We will, in the following denote $x^{\text{ref}}_{2,max}$ the number x_2 maximizing the number of successful preambles' transmissions. Resultantly, when the number of x_2 becomes higher, the number of RACH preambles fails increases and causes a heavier RACH congestion. This is due to RA reattempts but also to new arrivals after successful ACB checks. So, it will be interesting to design a brand new mechanism combined with an ACB scheme as a way to adapt dynamically the ACB factor to the congestion level. This may insure a number x_2 around $x^{\text{ref}}_{2,max}$ and as a result a maximum RA success probability.

Many research works investigated the dynamic aspect of ACB factor generation according to collision status. In [12], authors developed a Markov-Chain based traffic-load estimation scheme according to the collision levels. Then, they propose a spectrum of functions to control the barring factor and vary it with the estimated traffic load. In [13], with the fact that ACB and enhanced access barring (EAB) schemes both lack a contention detection method to decide when to activate it and how to adjust it properly, the authors proposed a context-aware dynamic resource allocation mechanism based on two-phases method: phase 1 for the estimation of random access attempts, and phase 2 for resource allocation. The authors in [14], proposed an estimation based adaptive ACB scheme to improve the scalability of M2M networks. To reach that, they first designed a new estimation method of the networks' loads and then proposed a function to adapt dynamically the ACB probability to network load conditions.

In our last contribution, we have proposed a novel method adapting dynamically, in real-time, the ACB factor according to the contention's level.

We will, in the following, detail the limitations of such a contribution and then expose our new proposal algorithm in order to correct such limitations.

III. Context

In a last contribution [15], we have proposed a dynamic approach to generate the appropriate ACB factor and adapt it dynamically according to overload situations. This is achieved using a Proportional Integral Derivative (PID) [16] controller that makes the number of M2M devices attempting the RA procedure converge to the optimal $x^{\text{ref}}_{2,max}$. The simulation results showed the efficiency of the PID controller in reducing RA collision probability as it avoids above the system overload and the radio resources' under-utilization.

However, in case of heavily congested M2M networks, this approach becomes insufficient to deal with such overloaded situations.

Simulation tests show that even if a PID regulator is applied, there are many situations, i.e. when the network load becomes heavier, where eNodeB is facing a synchronization problem. The latter is shown in Fig. 2. In fact, this leads to a rapid increase of the number x_2 synchronously coming at state x_2 (Fig. 2 (a)) and as a result the number of successful preambles transmissions intensively decrease. Such synchronized scenario has a higher potential to create a bursty and disruptive congestion. That's why, in some RACH opportunities, zero devices succeed the RA process (Fig. 2 (b)), as if it is a problem of compatibility between PID controller and the large number of M2M arrivals attempting ACB process.

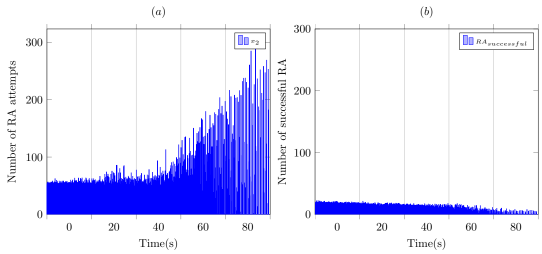

Fig. 2: Number of successful RA vs Number of RA attempts

We can also observe, from Fig. 3, that the obtained performance results (i.e. number of successful RA attempts) degrade when the network loads increases.

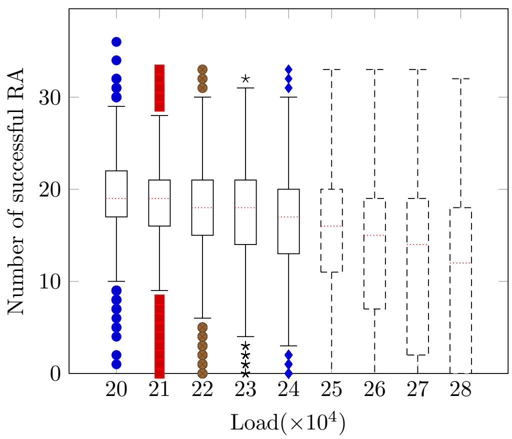

Fig. 3: Number of successful RA vs Load

Another way to prove the inefficiency of the PID approach to deal with extremely congested networks, can be achieved by comparing the evolution of resources' underutilization rate for various network loads. It can be easily seen, as shown in Fig. 4, that the resources' underutilization rate increases intensively when the network loads increases. This rate becomes intolerable ($\simeq 1$) when the network load reaches 25 $\times 10^4$.

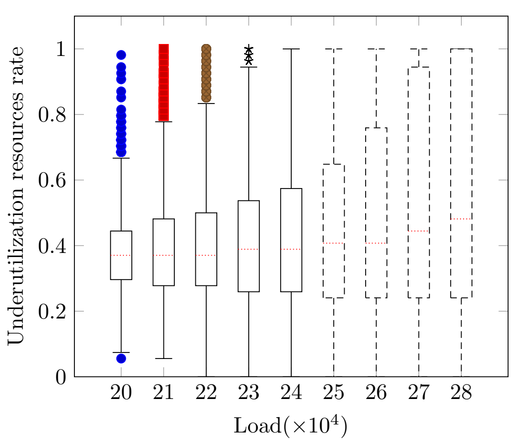

Fig. 4: Resource's under-utilization rate vs Load

From Fig. 5, we can also observe that when the network's load is higher, the number of abandons, i.e. the devices that leave the system after reaching the maximum RA preambles' re-transmissions, grows rapidly due to network load increase.

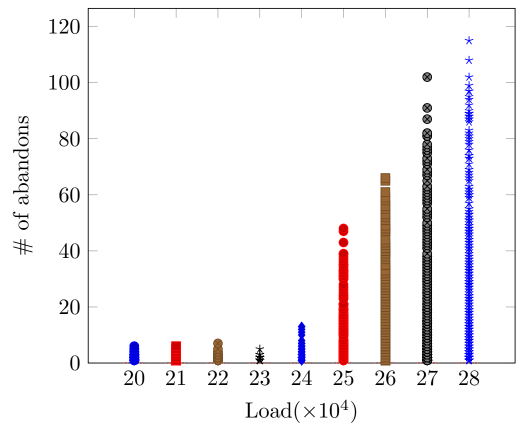

Fig. 5: Number of RA abandons vs Load

As a way to deal with such limitations, we propose a new approach that combines between PID controller and an improved dynamic method. The latter will be described in the following section.

IV. Proposal

A. DACB: Dynamic ACB algorithm

In this subsection, we will describe in details the proposed algorithm illustrated in the following block. Our algorithm consists of two phases performed at each RACH opportunity. In the following, $x_{2,max}^{\text{ref}}$ denotes the same fixed set point mentioned previously, and x_2^{ref} denotes a variable set point that will be used later.

In the first phase of DACB algorithm, eNodeB adjusts dynamically the set point x_2^{ref} according to the overload situation. It uses the Exponentially Weighted Moving Average (EWMA) values to check whether the overload level is high or not. If the congestion situations are heavy (line 6), eNodeB decreases x_2^{ref} as a way to forbid more devices from attempting access in the subsequent step. Otherwise, i.e. in case of relaxed overload conditions (line 9), eNodeB increases x_2^{ref} in order to allow more accesses in the following step. We consider only values within the interval $[0; x_{2,max}^{\text{ref}}]$ by applying line 11: $\min\left(\max\left(x_2^{\text{ref}}, 0\right), x_{2,max}^{\text{ref}}\right)$.

Once the dynamic adjustment's set point is performed, eNodeB executes the phase 2 to generate the ACB factor at step n. If the EWMA is less than $x_{2,max}^{\text{ref}}$ at step n, P_{acb} in the next step, is computed using equation in line 14, otherwise, eNodeB applied 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: $\min\left(\max\left(P_{acb}, 0\right), 1\right)$ (line 18).

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

Algorithm 1 DACB

```
$n \leftarrow 1$
$x_{2,max}^{\text{ref}} \leftarrow N$
$0 < \theta < 1, \alpha > 1, 0 < \beta < 1$
*Dynamic setpoint*:
$moy(n) \leftarrow (1-\theta)moy(n-1) + \theta x_2(n)$
**if** $moy(n) > \alpha x_{2,max}^{\text{ref}}$ **then**
    $x_2^{\text{ref}}(n+1) \leftarrow x_2^{\text{ref}}(n) - 1$
**else**
    **if** $moy(n) < \beta x_{2,max}^{\text{ref}}$ **then**
        $x_2^{\text{ref}}(n+1) \leftarrow x_2^{\text{ref}}(n) + 1$
$x_2^{\text{ref}}(n+1) \leftarrow \min\left(\max\left(x_2^{\text{ref}}(n+1), 0\right), x_{2,max}^{\text{ref}}\right)$
*ACB probability*:
**if** $moy(n) \leq x_{2,max}^{\text{ref}}$ **then**
    $P_{\text{acb}}(n+1) \leftarrow \frac{x_2^{\text{ref}}(n+1)}{x_1(n)}$
**else**
    $e(n+1) \leftarrow x_2^{\text{ref}}(n+1) - x_2(n)$
    $P_{\text{acb}}(n+1) \leftarrow K_{\text{p}}e(n+1) + K_{\text{i}}\sum_{k=0}^{n+1} e(k) + K_{\text{d}}(e(n+1) - e(n))$
$P_{\text{acb}}(n+1) \leftarrow \min\left(\max\left(P_{\text{acb}}(n+1), 0\right), 1\right)$
$n \leftarrow n+1$
**goto** *Dynamic set point*
```

B. RAES: RA attempts estimation algorithm

As eNodeB cannot have the information about the exact number of devices present in state x_2, we propose, in the following, to estimate it. Our estimation approach is illustrated in algorithm 2. First, eNodeB estimates the number of RACH attempts x_2 and then concludes the total number of ACB attempts x_1 at step n. During one RACH opportunity n, eNodeB can get the number of successful preambles $N_S(n)$ and 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 MTC devices, we can approximate x_2 using equation in line 6. By line 7, eNodeB computes the EWMA values of $\hat{x}_2$ at each step n. Then, it estimates the error $e(n)$ that reflects the distance between the measured $\hat{x}_2(n)$ and $x_{2,max}^{\text{ref}}$ (line 8). The estimation $\tilde{x}_2(n)$ is given by line 9.

Algorithm 2 RAES

```
1: $n \leftarrow 1$
2: $N_S, N_F$
3: $0 < \theta < 1$
4: $0 < \delta < 1$
5: $x_2$ *estimation*:
6: $\hat{x}_2(n) \leftarrow N_S(n) + 2N_F(n)$
7: $moy(n) \leftarrow (1-\theta)moy(n-1) + \theta\hat{x}_2(n)$
8: $e(n) \leftarrow e(n-1) + \delta(x_{2,max}^{\text{ref}} - \hat{x}_2(n))$
9: $\tilde{x}_2(n) \leftarrow moy(n) + e(n)$
10: $n \leftarrow n+1$
11: **goto** $x_2$ *estimation*
```

V. Performance evaluations

A. Simulation parameters

Having described in details our proposed algorithm, we direct now our focus to its performance using a discrete event simulator. Our simulator is developed under C and models the whole system described in section 2. During simulations, Poisson distribution is chosen as the MTC traffic model where inter-arrivals are exponentially distributed. We also adopt a RACH configuration where one RACH opportunity occurs every 10 ms with 54 preambles for each opportunity.

The simulations' parameters are summarized in Table I. When a RACH trial is declared unsuccessful, MTC device can retry the RA after a back off time chosen uniformly between 0 and the backoff parameter [6] fixed in the following table.

TABLE I: Simulation parameters

Parameters	**Values**
Simulation time	100s
N	54
R_{max}	10
Backoff parameter	20ms
ac-BarringTime	4s
Total number of M2M devices	28 $\times 10^4$

B. Simulation results

In this subsection, we first evaluate the performances of the proposed algorithm. To illustrate the effect of the DACB algorithm, we compare its performance against those obtained with a PID-based only method and a basic ACB method (fixed $p = 0.3$), for distinct network loads and different values of α and β. This comparison is made in terms of the following metrics: probability density function (PDF) of the total number of RA attempts, number of RA abandons, number of RA retransmissions, success probability, collision probability, idle probability of RA preambles and finally the average access delay. Here, we define the random access delay required by one device as the duration from the first RA attempt until the RA successful preamble transmission. Then, the average random access delay will be the ratio of the whole random access delay during one RACH opportunity to the total number of successful M2M devices in this opportunity.

Fig. 6 illustrates the evolution of the number of instantaneous and ewma values of RA attempts against the $x_{2,max}^{\text{ref}}$, when a DACB is applied. We can easily observe that even if the instant values oscillate between 40 and 70, the average ones remain very close to $x_{2,max}^{\text{ref}}$ which is one of the objectives of dynamic ACB generation as it helps to maximize the success access probability and minimize the resources' underutilization.

In Fig. 7 (a), we plot the PDFs of the number of RA attempts (i.e. x_2) against that of $x_{2,max}^{\text{ref}}$. We note that even we change the parameters α and β of DACB algorithm, the PDFs are maximized around $x_{2,max}^{\text{ref}}$, which proves the effectiveness of DACB. However, from Fig. 7 (b), we also observe that x_2 are spread far from $x_{2,max}^{\text{ref}}$. For example, when only a PID is applied, PDFs are maximized when $x_2 = 0$.

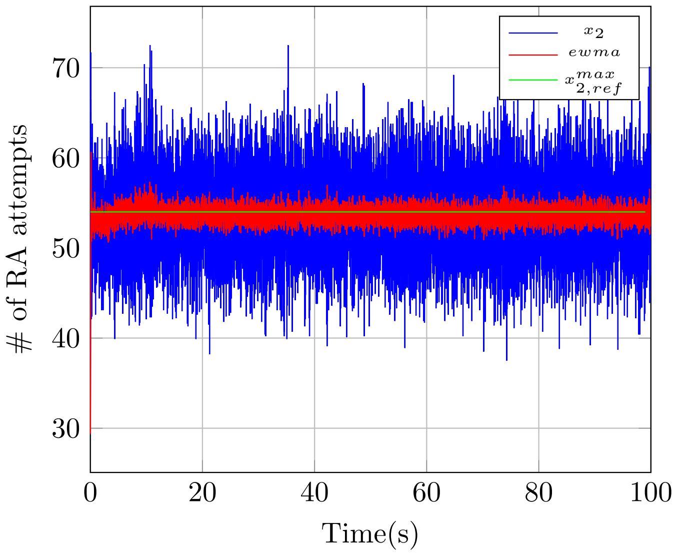

Fig. 6: Number of RA attempts vs Setpoint

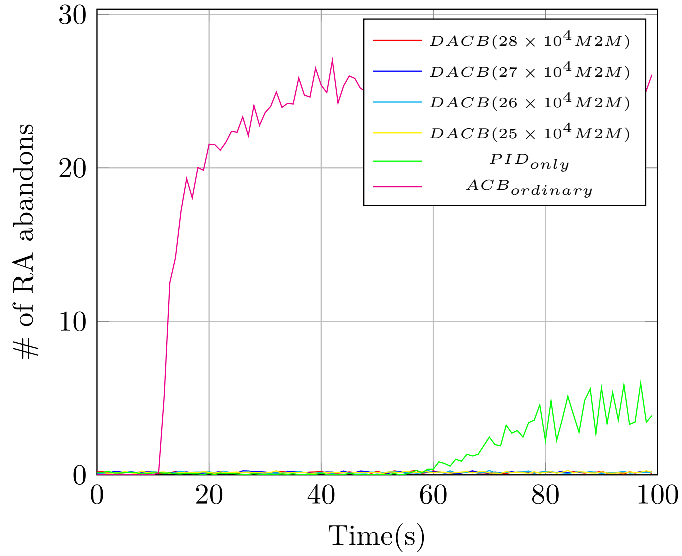

Fig. 8: Number of RA abandons

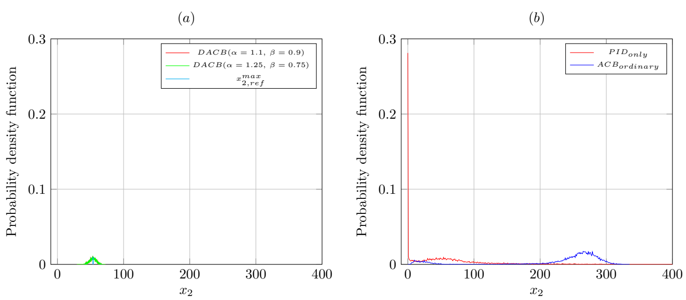

Fig. 7: Probability density function

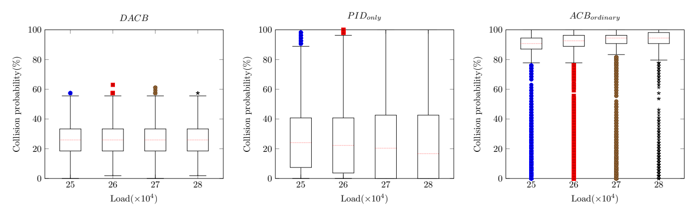

Fig. 9: Collision probability

Another way to test the performance improvements achieved by DACB can be reached by comparing the number of abandons for different network loads. Let's see the results obtained in Fig. 8. We first observe very little numbers of abandons (i.e. near to zero) and a greatly small fluctuation of these numbers even if the number of M2M devices increase. Whereas, in case of only PID-based method the number of abandons increases when the networks become more heavily congested (i.e. from $60s$). However, the number of abandons becomes intolerable from the beginning if an ordinary ACB scheme is applied.

From figure Fig. 9, we first note a very small fluctuation of collision probabilities between different network loads in case of DACB scheme. In fact, this is due to dynamic adjustment of the set point according to congestion level.Then, we can observe that collision probability is smaller when only a PID scheme is applied and even in case of very congested networks (i.e. 28×10^4 M2M). This is due to the synchronization problem observed in section 3. Another important observation is that collision phenomenon becomes exacerbated if we consider an ordinary ACB scheme.

On the other hand, we show the idle probability of preambles in Fig. 10. It can be easily seen that idle probability remains acceptable ($\simeq 40\%$), when a DACB is applied, even if the network loads increases, which leads to a maximum resources' utilization. However, idle probability becomes intolerable ($\simeq 100\%$) if an only PID based scheme is applied. This is in complete conformance with the synchronization problems discussed in section 3. We can also observe that idle probability decreases ($\simeq 0\%$) if an ordinary ACB scheme is considered. This is due to a large number of RA attempts (as seen in Fig. 7 (b)) and then the high collision probabilities obtained in Fig. 9.

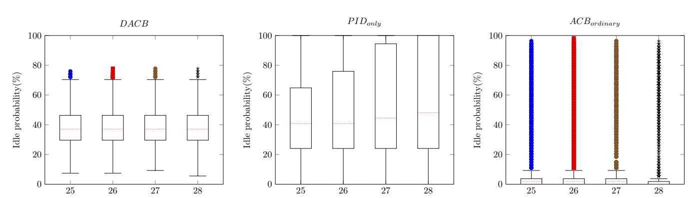

Fig. 10: Idle probability

We can now, from Fig. 11, notice the evolution of success access probability. We first observe that success probability is at its maximum ($\simeq 35\%$), with DACB algorithm, even if

M2M loads increases. Whereas, it decreases in case of only PID based scheme because of the synchronization problem discussed previously in section 3. Similar degradation results are obtained with an ordinary ACB scheme.

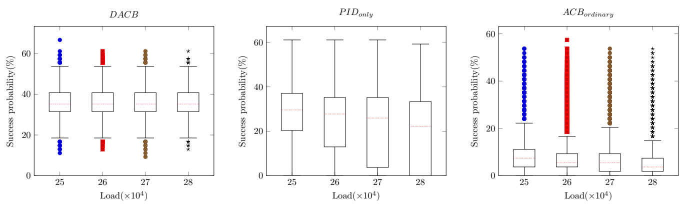

Fig. 11: Success probability

Fig. 12 shows the average number of RA preambles' re-transmissions. From the figure, we note that the mean number of re-transmissions is about 2.5 ($<< R_{max}$) when DACB scheme is adopted. However, this number exceeds 7 for PID-based method. Another important performance parameter is the average random access delay illustrated in Fig. 13. From Fig. 13 (a), we observe that the average delays for different loads are not much different and don't exceed $60ms$. On the other hand (in Fig. 13 (b)), we easily note that, with a PID-only strategy, delays are not tolerable. Nevertheless, even if the network is in relaxed conditions (i.e. 25×10^4 M2M), the average delay is higher than $100ms$.

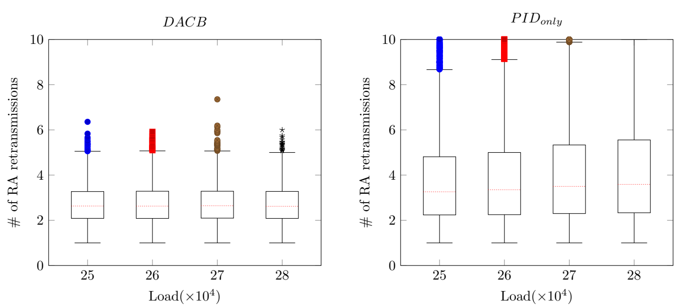

Fig. 12: Number of RA re-transmissions

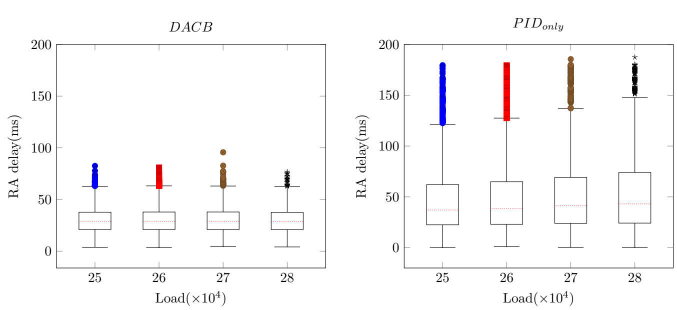

Fig. 13: Average random access delay

To evaluate the performances of the estimation scheme, we compare the actual number of RA attempts and the estimated one. The estimation simulation results are described in Fig. 14. There are some fluctuations between real values and predicted values, which remain acceptable.

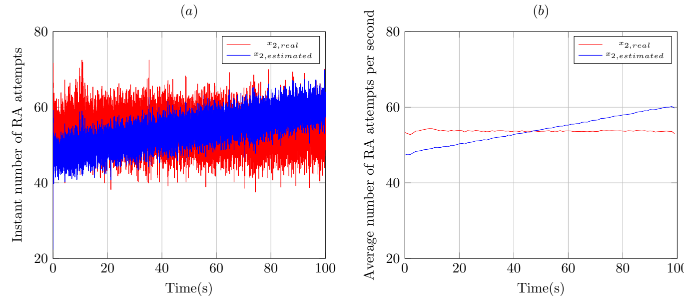

Fig. 14: RA attempts estimation

VI. Conclusions

In this paper, we addressed the problem of heavy congested M2M networks, which exhibit a risk of synchronized access of M2M devices, as showed in the paper.

A direct consequence of devices' synchronization is that an incremental increase of the number of M2M devices leads to either a significant reduction of the number of successful RA and resources' under-utilization, or to a congestion collapse, even when using an efficient ACB mechanism.

To cope with this problem, we have proposed the DACB mechanism, a new access control strategy for M2M random accesses. DACB represents a novel implementation of ACB method that consists in: (i) an accurate estimation of the network load, (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) calculating the ACB probability according to the expected network's overload situations.

The simulation results demonstrated the efficiency of the DACB algorithm in terms of reduced congestion probability and maximized success access probability, compared to existing approaches. In fact, DACB outperforms clearly both the classical ACB method and the PID-based one, which is known to be very efficient. Additionally, the results showed a minimized number of RA abandons as well as 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.

We direct, as future works, our research to M2M energy aspects. Indeed, it will be very interesting to know how much DACB can improve the MTC device's battery consumption.

References

[1] 3GPP TR 23.888, *Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications*, V.11.0.0, Septembre 2012.

[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] 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.

[4] 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.

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

[6] 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.

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

[8] 3GPP TR 37.868, *RAN Improvements for Machine-type Communications*, V.11.0.0, March 2013.

[9] 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.

[10] 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.

[11] 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.

[12] 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. 12

[13] Y. Pang, G. Lin and H. Wei, *Context-aware Dynamic Resource Allocation for Cellular M2M Communications*, Internet of things journal, 2015. 13

[14] 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.

[15] 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.

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