2016 IEEE Symposium on Computers and Communication (ISCC)

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 in addition to terminals' synchronization issues. In this paper, we proposed a novel implementation of the Access Class Barring (ACB) scheme, 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 collapse, dynamic ACB factor.

I. Introduction

Machine-to-Machine (M2M) communications, with their capability of providing various types of applications and services [1], are already considered to be a crucial technology for 4G LTE-Advanced systems. They are also expected to become a key enabler for the fifth generation (5G) networks being part of Internet of things (IoT) [2][3].

Indeed, 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) [4]. Therefore, this numerous number of M2M devices may arrive simultaneously asking for network connection establishment [2].

To achieve that purpose, every device should first perform the RACH (Random access channel) procedure [5] [6]. 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 tackle the problem of congestion at its root, 3GPP introduced the Access Class Barring (ACB) technique [7]. In fact, as a way to avoid useless M2M devices connections' attempts, the ACB mechanism defines both a probability p, called the *ACBfactor*, and a time-duration, called the $Tbarring$ [7]. Many contributions in the literature showed that generating a dynamic value for the ACB factor according to the network overload is very efficient [8][9][10]. 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, which results in poor performances.

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.

The remainder of this article is organized as follows. Section 2 portrays our new adaptive access proposed protocol. Section 3 is dedicated to the simulation setup and the discussion of the obtained results. Finally, the paper concludes in Section 4 with a summary recapping the main advantages and achievements of the proposed access protocol.

II. DACB: Dynamic ACB for heavily congested M2M networks

In this section, we will describe in details the proposed algorithm, named DACB, which is performed at each RACH opportunity.

In the envisioned scenario, a pool of x_1 devices goes to a first state x_1 and starts the ACB check. Once the ACB check is successful, the M2M devices move to the state x_2 to attempt the RA. x_2 also depicted the total number of devices present at state x_2. We will, in the following, denote $x_{2,max}^{\text{ref}}$ as the number x_2 maximizing the number of successful preambles' transmissions (i.e. targeted load) [11] while x_2^{ref} denotes a variable set point that is derived dynamically.

In the first phase of the DACB algorithm, the eNodeB adjusts dynamically the set point x_2^{ref} according to the overload situation. It uses the Exponential Weight Moving Average (EWMA) values Ave of x_2 to check whether the overload level is high or not. If the congestion situation is heavy (line 6), the eNodeB decreases the x_2^{ref} value, which enables to block more devices from attempting the random access in the subsequent step. Otherwise, i.e. in case of relaxed overload conditions (line 9), eNodeB increases the x_2^{ref} in order to allow more devices to attempt the access in the following step. We consider only values within the interval $[0; x_{2,max}^{\text{ref}}]$.

Once the dynamic targeted load determined, the eNodeB executes the phase 2 to generate the ACB factor at step n. If the average is less than $x_{2,max}^{\text{ref}}$ at step n, the P_{acb}, in the next step, is computed using equation in line 14. Otherwise, the 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

1: $n \leftarrow 1$
2: $x_{2,max}^{\text{ref}} \leftarrow N$
3: $0 < \theta < 1, \alpha > 1, 0 < \beta < 1$
4: *loop*:
5: $Ave(n) \leftarrow (1-\theta)Ave(n-1) + \theta x_2(n)$
6: **if** $Ave(n) > \alpha x_{2,max}^{\text{ref}}$ **then**
7: $\quad x_2^{\text{ref}}(n+1) \leftarrow x_2^{\text{ref}}(n) - 1$
8: **else**
9: $\quad$ **if** $Ave(n) < \beta x_{2,max}^{\text{ref}}$ **then**
10: $\quad\quad x_2^{\text{ref}}(n+1) \leftarrow x_2^{\text{ref}}(n) + 1$
11: $x_2^{\text{ref}}(n+1) \leftarrow \min\left(\max\left(x_2^{\text{ref}}(n+1), 0\right), x_{2,max}^{\text{ref}}\right)$
12: **if** $Ave(n) \leq x_{2,max}^{\text{ref}}$ **then**
13: $\quad P_{\text{acb}}(n+1) \leftarrow \frac{x_2^{\text{ref}}(n+1)}{x_1(n)}$
14: **else**
15: $\quad e(n+1) \leftarrow x_2^{\text{ref}}(n+1) - x_2(n)$
16: $\quad P_{\text{acb}}(n+1) \leftarrow K_{\text{p}} e(n+1) + K_{\text{i}} \sum_{k=0}^{n+1} e(k) + K_{\text{d}}(e(n+1) - e(n))$
17: $P_{\text{acb}}(n+1) \leftarrow \min\left(\max\left(P_{\text{acb}}(n+1), 0\right), 1\right)$
18: $n \leftarrow n+1$
19: **goto** *loop*

III. 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 for 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 [5] 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	28×10^4

B. Simulation results

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. This comparison is made in terms of the following metrics: number of RA abandons, collision probability and finally idle probability of RA preambles.

Fig. 1 illustrates the evolution of the number of instantaneous and average 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.

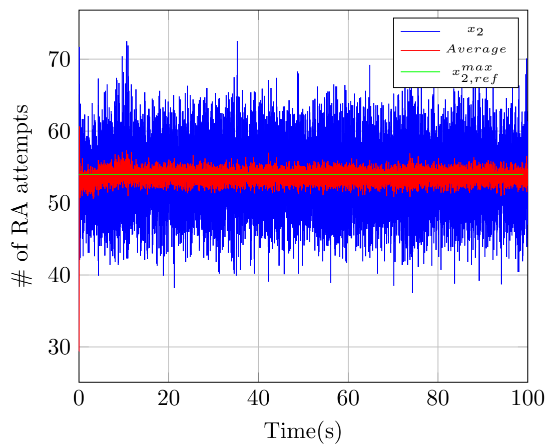

Fig. 1: Number of RA attempts vs Setpoint

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. 2. 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 increases. 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 is intolerable from the beginning when applying an ordinary ACB scheme.

From Fig. 3, 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 the dynamic adjustment of the set point according to congestion level. Then, we can observe that collision probability is smaller when 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 introduced in this paper. Another important observation is that the collision phenomenon becomes exacerbated when considering an ordinary ACB scheme.

On the other hand, we show the idle probability of preambles in Fig. 4. It can be easily seen that idle probability remains acceptable ($\simeq 40\%$), when a DACB is applied, even

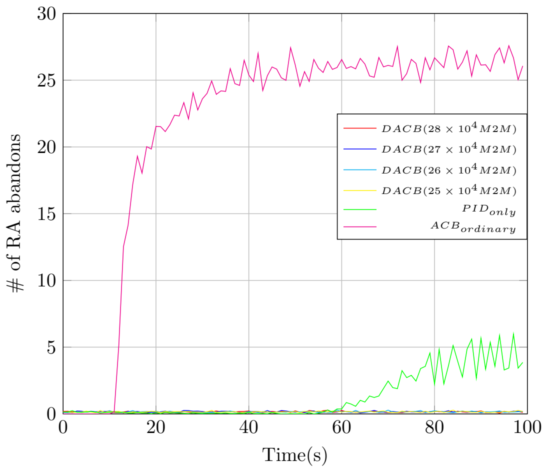

Fig. 2: Number of RA abandons

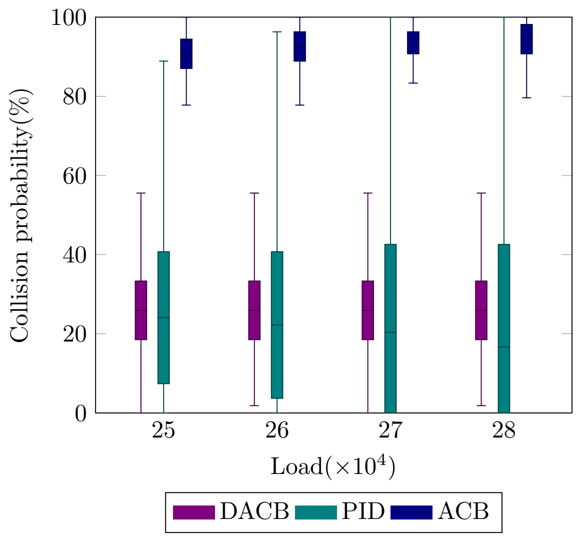

Fig. 3: Collision probability

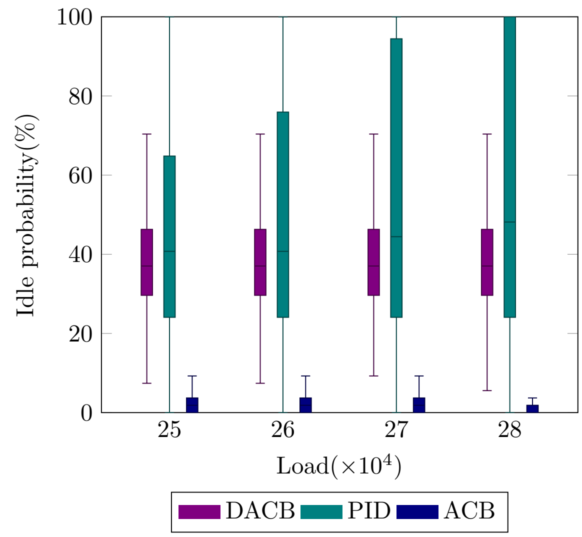

Fig. 4: Idle probability

if the network loads increases, which leads to a maximized resources' utilization. However, idle probability is intolerable ($\simeq 100\%$) when applying the PID scheme. This is in complete conformance with the synchronization problems discussed above. 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 and then the high collision probabilities obtained in Fig. 3.

IV. Conclusions

In this paper, we addressed the problem of heavy congested M2M networks, which exhibit a risk of synchronized access of M2M devices. A direct consequence of devices' synchronization is either a significant reduction of the number of successful RA and resources' under-utilization, or a congestion collapse, even when using an efficient ACB mechanism.

To cope with this problem, we have proposed a novel implementation of the ACB method, DACB, that consists in: (i) a dynamic adjustment of the model's parameters depending on RA congestion level, and (ii) 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, maximized success access probability and a minimized number of RA abandon, compared to existing approaches.

As a future work, we can direct our research to DACB impact on the M2M device's battery consumption.

References

[1] 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.

[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] 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.

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

[7] 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.

[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] 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.

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

[11] Bouzouita M, Hadjadj-Aoul Y, Zangar N, Rubino G and Tabbane S, *Applying nonlinear optimal control strategy for the access management of MTC devices*, IEEE/CCNC, Las Vegas, 2015.