



HAL
open science

Estimating the number of contending IoT devices in 5G networks: Revealing the invisible

Meriam Bouzouita, Yassine Hadjadj-Aoul, Nawel Zangar, Gerardo Rubino

► To cite this version:

Meriam Bouzouita, Yassine Hadjadj-Aoul, Nawel Zangar, Gerardo Rubino. Estimating the number of contending IoT devices in 5G networks: Revealing the invisible. Transactions on emerging telecommunications technologies, 2018, e3513, pp.1-18. 10.1002/ett.3513 . hal-01933968

HAL Id: hal-01933968

<https://inria.hal.science/hal-01933968>

Submitted on 24 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

JOURNAL ARTICLE

Estimating the number of contending IoT devices in 5G networks: revealing the invisible

Meriam Bouzouita^{‡1,2} | Yassine Hadjadj-Aoul^{§1} | Nawel Zangar^{†2} | Gerardo Rubino^{#3}¹University of Rennes 1, France²Mediatron, SUP'COM, Tunisia³INRIA, Rennes, France**Correspondence**[‡]Email: mariem.bouzouita@supcom.tn,[§]Email: yhadjadj@irisa.fr,[†]Email: nawel.zangar@insat.rnu.tn,[#]Email: Gerardo.Rubino@inria.fr.**Summary**

The explosive growth of connected objects is certainly one of the most important challenges facing operators' network infrastructures. Although it has been foreseen for a very long time, it is still not clear how to support such huge number of devices efficiently. A smarter planning of dedicated access slots would certainly limit the burden but remains insufficient since some equipments react to events which cannot be timed. Moreover, barring some IoT devices from accessing the network is very efficient; nevertheless, efficiency is generally linked to precise knowledge of the number of contending devices. Though, before connection establishment, the terminals are invisible to access points and, therefore, it is very difficult to estimate their number. A lower bound of backlogged devices can be determined. However, underestimating this number may lead to a congestion collapse whereas an overestimation implies underutilization of resources. In this paper, we propose a lightweight change to the standard to accurately reveal the state of network congestion by overloading connections' requests with the number of access attempts (number of times the device has been barred as well as number of attempts). Using such information, we propose an accurate recursive estimator of the number of devices. The obtained results demonstrated that the proposed solution not only makes it possible to estimate the number of equipments much better than existing techniques, but also allows determining precisely the number of blocked equipments.

KEYWORDS:

Massive accesses, Multi-class, IoT devices, Access Class Barring, estimation, network's overload.

1 | INTRODUCTION

The fast development of radio communications has led to a larger bandwidth demand putting more and more pressure on mobile network's operators (NO). As a consequence, NO are adapting to these changes by proposing the new standard 5G for mobile networks, which envisions to improve support for existing and future services in a more flexible and agile way. In this way, 5G¹ will include not only improvements at the core network, which tends to be more virtualized, but also at the access network, which will undergo a massive arrival of the Internet of Things (IoT), behind which there are huge expectations.

The excessive growth of the number of IoT² devices represents a real challenge, that needs to be taken into account by NOs, at the risk otherwise of deteriorating all other services. Indeed, the limited radio resources, the random access to the medium

and the spectrum sharing between different applications are critical issues in the IoT context³. Therefore, radio networks need to be optimized to enable massive access of low cost and energy-efficient IoT devices⁴.

Low Power Wide Area Networks⁵ (LPWAN) were developed to meet all IoT requirements, since their main aims consist in ensuring energy-efficient, simple and low cost devices communicating in long range at a low bit rates. Many solutions emerged, in particular, non-cellular IoTs, operating in unlicensed bands, such as LoRa and SigFox⁶. These technologies have a fast deployment process focusing more on energy consumption, very long range and low data rate. Nevertheless, the MAC nature of such proprietary LPWAN technologies, i.e. ALOHA-based access, is inefficient to face a massive access of IoT devices⁷.

More recently, the Third Generation Partnership Project (3GPP) adapted its cellular network to meet the IoT specifications and needs. In release 13, a software upgrade of existing networks is provided to support the newly introduced Narrowband IoT^{8,9,10}(NB-IoT) and the Extended Coverage GSM (EC-GSM) as cellular IoT solutions. 3GPP considers RAN overload control as the first priority improvement area; it introduced several overload resolution mechanisms in order to tackle the explosive growth of simultaneous IoT connections¹¹. Hence, several Physical RACH (PRACH) overload resolution methods were defined to improve the support of such devices in LTE-Advanced networks. First, Access Class Barring (ACB) mechanism was defined by 3GPP in Release 8 as an access barring solution for all IoT transmission and an Extended Access Barring, EAB, was defined in release 11 aiming to better prevent RAN overload¹².

Many implementations of ACB and EAB were introduced in the literature to avoid congestion at the network's access^{13,14,15}. Nevertheless, these solutions fail to avoid a synchronized access of IoT devices. Indeed, since the eNodeB ignores the number of IoT devices willing to connect, it cannot predict accurately the overload status, which may lead to a congestion collapse.

In this regard, we focus our research work on improving the NB-IoT network by investigating the Random Access Channel (RACH) for massive IoT accesses. We consider two main states, the number of backlogged devices waiting for EAB check and the total number of devices that pass the EAB check and wait for the random access (RA). As the number of contending devices is invisible until connection establishment, it is very hard to estimate it. In this paper, we propose new extensions and substantial improvements of our previous work. They are summarized in the following points:

- we offer a detailed and up-to-date state of the art;
- we extend our previous simple fluid model¹⁶, which represents only one class of traffic, to study the access of multi-class IoT devices into a mobile network;
- we refine our initial estimate¹⁶ of the number of terminals, from the different classes of IoT traffic, that have attempted the access;
- we propose an estimation of the number of devices of different classes willing to connect. To the best of our knowledge, this is the first work in the literature addressing this issue.

The most significant novelty of this paper is to propose an accurate and insightful recursive estimation technique of the number of contending devices for each class. Based on the radio resources in terms of successful, collided or idle preambles, we can deduce the average number of terminals that attempted the access and finally we make a step backward to adjust the estimation of the backlogged terminals, waiting for the EAB check. The main specificity of our approach consists in leveraging a crowd-sourced information. In fact, we propose to enrich each connection request with the total number of EAB and RA attempts. Thus, we introduce a lightweight change to the standard to allow the eNodeB to be notified about the congestion level.

The integration of the proposed approach will considerably reduce access costs by efficiently supporting a very massive number of terminals, which will allow to go much farther than the 50000 terminals per cell expected by the current standard. Indeed, the more accurate the estimate of the number of terminals, the more effective their access control will be.

The remainder of this paper is organized as follows. Section 2 is dedicated to some related works. Section 3 portrays our proposed model of access for multi-class IoT devices and studies its controllability. Section 4 describes the proposed algorithms given to estimate the number of terminals that attempted the access and those wanting to access. 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 model.

2 | RELATED WORKS

In order to support an increasing number of IoT devices, networks operators and service providers will have to use different wireless access technologies. A large share of IoT devices will be served by short-range radio technologies operating on unlicensed spectrum (e.g. Wi-Fi and Bluetooth). For these networks, several challenges will need to be addressed in order to make these networks viable solutions for IoT support¹⁷. Among these, routing¹⁸, providing the requested Quality of Service (QoS)/Quality of Experience (QoE) support, fault tolerance remains open problems, which has been the subject of several works in the literature.

A significant proportion of IoT devices will still be enabled by wireless Wide Area Networks (WANs). These networks have the advantage of covering large areas at low cost. On the one hand, LPWAN networks, operating in unlicensed frequency bands such as LoRa or Sigfox, which have been dedicated to IoT devices, are being deployed on a large scale and provide connectivity to different types of IoT devices. On the other hand, 3GPP networks, generally operating on licensed frequencies, are already widely available worldwide and can offer a reliable connectivity service at a reasonable cost, through notably NB-IoT.

Massive access of IoT devices through wireless wide access networks, and particularly LPWAN, represents one of the most important challenges that network operators will face. Indeed, the intrinsic limits in radio resources, the random access of IoT terminals, whose number continues to grow, presents a real risk of congestion collapse¹⁹, which calls for the development of flexible and efficient scheduling and random access techniques²⁰.

The fact is, not all LPWAN networks are equal regarding the support of massive IoT devices. Although released much earlier, LoRaWan and Sigfox do not propose a solution to this problem, apart from the limitation of the “Time On Air” (TOA)⁷. TOA allows reducing the burden at the access but fails in adapting to network conditions’ variation. Hence, the optimization of the capacity of LoRaWAN and Sigfox remains an important issue to deal with. On the other hand, RAN overload control was identified very early as a first priority improvement area in 3GPP¹¹. The European Telecommunications Standards Institute (ETSI) also introduced several solutions to overcome such problems in mobile networks²¹. Thus, the newly introduced standards for IoT, such as the NB-IoT, benefit naturally from these advances. Among the concepts introduced, the Extended Access class Barring (EAB) technique is certainly the one that attacks the problem at its root, since it allows blocking access attempts for IoT terminals²².

To combat the overlapped simultaneous transmissions of multiple devices in a shared radio resource, a non-orthogonal multiple access (NOMA)^{23 24} strategy is proposed, where multiple devices are allowed to transmit over the same sub-band and the base station performs successive interference cancellation to decode each device’s message. The authors derived the system stability conditions, where the maximum packet arrival rate was found with and without quality of service guarantee and the obtained results allows to clearly improve the NB-IoT strategy²⁵.

In this paper, we focus on the EAB technique, which can be used in conjunction with the techniques introduced above. Indeed, the EAB proves its efficiency and remains necessary when an access of massive number of IoT devices is required. The idea introduced is rather simple, since it involves calculating a blocking factor. However, a good implementation would require a good knowledge of the number of terminals willing to attempt access to infer the probability of blocking. Unfortunately, this information is not available in the network. Thus, approaches based on an exact knowledge of the system model make it possible to obtain an optimal controller, as the one introduced in our previous work²⁶, but are generally not operational in practical situations. This explains the proliferation of system model-agnostic approaches²⁷.

Most of existing approaches are based on estimating the number of terminals that attempted access, as they cannot have an accurate knowledge of the system’s status. Thus, the number of IoT devices that succeeded at the access, knowing the probability of blocking, makes it possible to have an indicator on the number of terminals that wanted to attempt access. In this, we proposed a dynamic adaptive access barring scheme for heavily congested M2M networks¹⁶. The main idea behind the paper is to, first, determine the optimal number of devices which should attempt access. Then, to propose a system model-agnostic controller, which adapts the blocking probability based on the mismatch existing between the calculated and the targeted number of successes (i.e. optimal one). The obtained results demonstrated a reduced random access delay in addition to reduced re-transmissions, which has a direct impact on the IoT energy consumption. Nevertheless, no traffic differentiation is considered and all IoT devices are treated identically. Using a similar approach for deducing the optimal load, the authors, in previous works^{28 29}, proposed to rely on the calculation of the steady state probabilities of a Markov-chain to deduce an efficient control scheme. A similar method was proposed by Yuan-Chi Pang et al³⁰. This method estimates the number of random access attempts, and then proposed a scheme for resources’ allocation. The proposed solution is applicable in many IoT, as it insures a high resource efficiency and low random access delay without a prior knowledge of arrival traffic rate. To offer an individual reliability guarantee per device, another proactive approach³¹ is proposed. It consists of two phases, an estimation phase and a

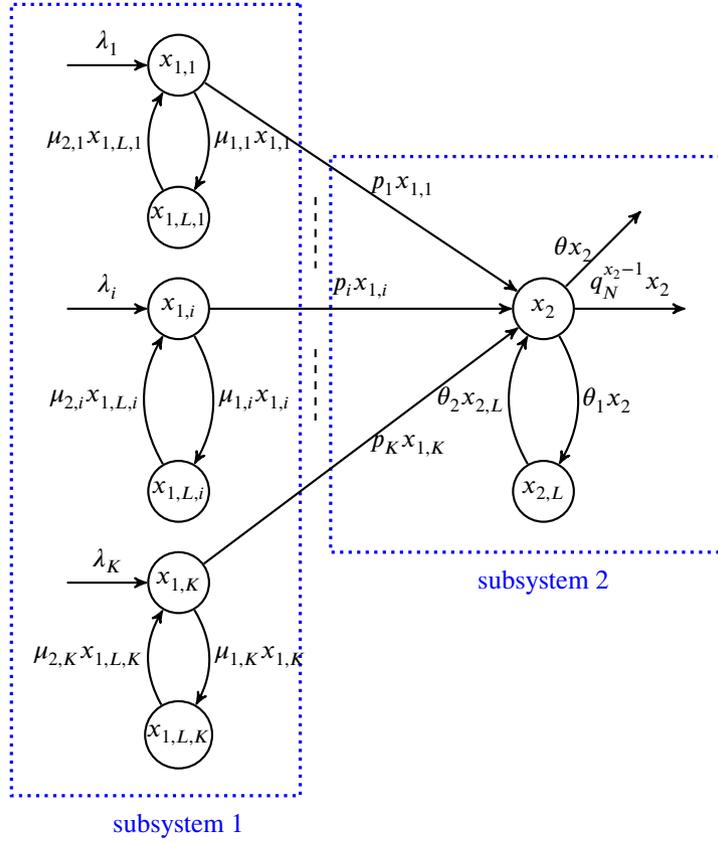


FIGURE 1 System model. Subsystem 1 represents the terminals, belonging to different classes (from 1 to K), that would like to connect; the objects in the state variable $x_{1,i}$ represent those that can try to connect with a probability p_i , in the case of a failure they go into the waiting state $x_{1,L,i}$ for a back-off time duration. Subsystem 2 represents the objects coming from the different classes that can try to choose a preamble. In the case of a collision, they may attempt access a number of times. They leave subsystem 2 when they succeed in being the only ones to have chosen a preamble or when they reach the maximum number of attempts (with a rate of θ).

serving phase. In the estimation phase, the number of arrivals is estimated and then this information is used to tune the amount of resources allocated in the serving phase.

While efficient and powerful, these techniques fail to avoid a synchronized access for massive M2M devices¹⁹, which may result in some cases to a congestion collapse. On the other hand, only few approaches in the literature proposed to rely on crowd sourcing to estimate the system model state. Without being exhaustive, a crowd sourcing algorithm is proposed to adjust dynamically the back-off times according to congestion level and the traffic's type³².

In this way, we propose in this paper to reveal the number of IoT devices willing to connect through crowd sourcing. To the best of our knowledge no one proposed an accurate estimation of such number, especially by class of IoT applications, which emphasizes the originality of our work.

3 | A MODEL OF ACCESS FOR MULTI-CLASS IOT DEVICES

3.1 | Model's definition

Our model for IoT devices' random access with multiple EAB factors (i.e. one factor per class) is influenced by the single-class system model that we have proposed in our previous work¹⁶.

The proposed model is a fluid one: the involved quantities and the whole numbers are seen as real (continuous) quantities. The parameters used are listed below:

- $x_{1,i}(t)$ number of backlogged devices from class i at time t , where $i \in \mathbb{C} = \{1, 2, \dots, K\}$, K thus being the number of considered classes;
- $x_{1,L,i}(t)$ number of blocked devices from class i waiting for a re-attempt at time t , after having failed an EAB check;
- $x_2(t)$ total number of devices from the different classes that pass the EAB check and wait to start RA attempt at time t ;
- λ_i arrival rate of devices from class i . Different traffic patterns will be considered in the following, depending on the type of IoT applications;
- $\mu_{1,i}$ rate of EAB failure for class i , which is equal to $1 - p_i$;
- $\mu_{2,i}$ rate of EAB re-attempts for class i ;
- θ_1 rate of RA failure, which is equal to $1 - q_N^{x_2-1}$ when θ is equal to 0 (see last item);
- θ_2 rate of RA re-attempts;
- θ rate at which the devices abort the transmission after reaching the maximum number of RA attempts; in a correctly dimensioned system, we should have $\theta = 0$;
- p_i EAB factor for class i .

When attempting the random access, IoT devices contend for the same available preambles. As depicted by the 3GPP standard, the number of preambles N should be an integer in the interval [4..64].

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 compute the average values of these quantities that we have determined in another paper³³. These quantities will be used by our 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 (this is a classic 'balls into bins' problem):

$$N_S = \sum_{i=1}^K N_S^i = q_N^{x_2-1} x_2, \quad (1)$$

where N_S^i represents the number of successful preambles chosen from class i .

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

$$N_I = N q_N^{x_2}. \quad (2)$$

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

$$N_F = N - (N_S + N_I). \quad (3)$$

Now we are ready to describe the evolution of the state variables $x_{1,i}(t)$, $x_{1,L,i}(t)$, $x_2(t)$ and $x_{2,L}(t)$, for all $i \in \mathbb{C}$, based on the model depicted in Fig. 1. The model's dynamics is described by the following system of differential equations: for all i ,

$$\left\{ \begin{array}{l} \frac{dx_{1,i}}{dt} = \lambda_i - p_i x_{1,i} - \mu_{1,i} x_{1,i} + \mu_{2,i} x_{1,L,i}, \\ \frac{dx_{1,L,i}}{dt} = \mu_{1,i} x_{1,i} - \mu_{2,i} x_{1,L,i}, \\ \frac{dx_2}{dt} = \sum_{i=1}^K p_i x_{1,i} + \theta_2 x_{2,L} - (\theta + \theta_1) x_2 - x_2 q_N^{x_2-1}, \\ \frac{dx_{2,L}}{dt} = \theta_1 x_2 - \theta_2 x_{2,L}. \end{array} \right. \quad (4)$$

with the constraints given below:

- $x_{1,i}$, $x_{1,L,i}$, x_2 and $x_{2,L}$ should be nonnegative for all i ,
- $\lambda_i > 0$, $\theta_1 > 0$, $\theta_2 > 0$, $\mu_{1,i} > 0$, $\mu_{2,i} > 0$ and $\theta \geq 0$ for all i ,
- $0 \leq p_i \leq 1$ for all i .

3.2 | Steady state analysis

In this section, we direct our focus on the analysis of its steady state performance.

For the sake of simplicity, we consider in the following the case where θ is equal to 0. Indeed, in a system working properly, this assumption should be valid.

The study of the steady state of the system comes back to consider that the derivatives expressed in equations (4–7) are equal to 0.

$$\begin{cases} 0 &= \lambda_i - p_i \bar{x}_{1,i} - (1 - p_i) \bar{x}_{1,i} + \mu_{2,i} \bar{x}_{1,L,i}, & (8) \\ 0 &= (1 - p_i) \bar{x}_{1,i} - \mu_{2,i} \bar{x}_{1,L,i}, & (9) \\ 0 &= \sum_{i=1}^K p_i \bar{x}_{1,i} + \theta_2 \bar{x}_{2,L} - (1 - q_N^{\bar{x}_2 - 1}) \bar{x}_2 - \bar{x}_2 q_N^{\bar{x}_2 - 1}, & (10) \\ 0 &= (1 - q_N^{\bar{x}_2 - 1}) \bar{x}_2 - \theta_2 \bar{x}_{2,L}. & (11) \end{cases}$$

From (8) and (9), we obtain, for all i ,

$$\bar{x}_{1,i} = \frac{\lambda_i}{p_i}. \quad (12)$$

Replacing the value of $\bar{x}_{1,i}$ in (9), we have

$$\bar{x}_{1,L,i} = \frac{1 - p_i}{p_i} \frac{\lambda_i}{\mu_{2,i}}. \quad (13)$$

By adding equations (10) and (11) we have

$$-\bar{x}_2 q_N^{\bar{x}_2 - 1} + \sum_{i=1}^K \lambda_i = 0. \quad (14)$$

Write $\lambda = \sum_{i=1}^K \lambda_i$. We have

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

The numerical solution to this equation in \bar{x}_2 can be easily obtained following a fixed-point approach (see next subsection), by iterating the sequence $(s_n)_{n \geq 0}$, with $s_0 = \lambda$ and

$$s_{n+1} = s_n (1 - q_N^{s_n - 1}) + \lambda. \quad (15)$$

The convergence is very fast (a few iterations provide a very precise value). We can also note that the solution to (14) can be written in terms of the Lambert W function, defined by $W(x)e^{W(x)} = x$ ³⁴. This function is available in most scientific libraries and numerical analysis tools. We have

$$\bar{x}_2 = W(q_N \lambda). \quad (16)$$

Knowing \bar{x}_2 , the last component of the unique stationary point follows from (11):

$$\bar{x}_{2,L} = \frac{1}{\theta_2} (1 - q_N^{\bar{x}_2 - 1}) \bar{x}_2. \quad (17)$$

So, the stationary or equilibrium point of the dynamical system is given by

$$\begin{aligned} \bar{x}_{1,i} &= \frac{\lambda_i}{p_i}, \\ \bar{x}_{1,L,i} &= \frac{1 - p_i}{p_i} \frac{\lambda_i}{\mu_{2,i}}, \\ \bar{x}_2 &= W(q_N \lambda), \\ \bar{x}_{2,L} &= \frac{1}{\theta_2} (1 - q_N^{\bar{x}_2 - 1}) \bar{x}_2. \end{aligned}$$

3.3 | Optimal number of IoT devices

Having too many contending IoT devices causes irretrievably collisions and, therefore, an inefficient access; having too few devices would result in under-utilization of resources. The main idea, in this section, is to derive an optimal number of contending devices x_2 , which maximizes the success access probability.

All the devices passing successfully the EAB check will contend for the same N radio resources to access the network. The optimal number of devices x_2^* , from the different classes, performing an RA at the same time is the number x_2 maximizing N_S . To compute it, we propose to analyze the nonlinear equation given in 1). We provide here, a short summary following a direct approach, avoiding the classic use of the Lambert function W defined on any complex number z by $W(z) = w \iff z = we^w$ ³⁴ as in previous subsection.

Let's us define 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 $-1/(eq \ln q)$. Replacing q by $q_N = 1 - N^{-1}$ and x by \bar{x}_2 , this already gives us the stability condition of the "right side" of the model, that is, concerning the convergence of $x_2(t)$ and $x_{2,L}(t)$: $\lambda \leq -1/(eq_N \ln q_N) = N_S^*$. If $\lambda < N_S^*$, then we have two solutions to the equation $\lambda = \bar{x}_2 q_N^{\bar{x}_2-1}$, say r_a and r_b , with $r_a < -1/\ln q_N = \bar{x}_2^* < r_b$, where r_a leads to stability and r_b to instability (see Fig. 2). Knowing that $0 < r_a < \bar{x}_2^*$ allows to easily find r_a numerically, for instance using a Newton scheme³⁵, or using a fixed-point method as in previous subsection. Let us provide a bit more details here. We defined the sequence $(s_n)_{n \geq 0}$ by $s_0 = \lambda$ and $s_{n+1} = \lambda/q_N^{s_n-1}$. Write back q instead of q_N to simplify the presentation. We can first check by induction that this sequence is increasing. At the beginning of the sequence, $s_1 > \lambda = s_0$ because $q^y < 1$ for all $y > 0$. Then, assuming that $s_0 < s_1 < \dots < s_n$, we must prove that $s_n < s_{n+1}$, that is, that

$$\frac{\lambda}{q^{s_{n-1}-1}} < \frac{\lambda}{q^{s_n-1}} \iff q^{s_n-1} < q^{s_{n-1}-1},$$

which is true because function $x \mapsto q^{x-1}$ is decreasing and $s_{n-1} < s_n$. So, $(s_n)_{n \geq 0}$ is strictly increasing.

Then, we can check that the sequence is upper-bounded, for instance by x^* . First, for s_0 ,

$$s_0 = \lambda < \frac{\lambda}{q^{x^*-1}},$$

and since $x^* q^{x^*-1} = x^*/(eq)$, then $q^{x^*-1} = 1/(eq)$ and we have

$$s_0 < \frac{\lambda}{q^{x^*-1}} = eq\lambda < x^*.$$

Then, assuming that $s_n < x^*$, we have

$$s_{n+1} = \frac{\lambda}{q^{s_n-1}} < \frac{\lambda}{q^{x^*-1}}.$$

Now, we conclude as we did for s_0 . This proves that when $n \rightarrow \infty$, $s_n \uparrow r_a$; as we stated before, the convergence is very fast.

Concerning the "left" part of the model (that is, the variables indexed by 1), the stationary point is always stable. This can be checked by any standard technique. Intuitively, the dynamics is similar as that of x in the ODE $x' = a - bx$, with $a, b > 0$: for any $x(0)$, a, b , we have $x \rightarrow a/b$ and the equilibrium is stable. One can even solve explicitly that part of the differential system (for instance, using Maple or Mathematica) and observe that the solutions are continuous in the initial conditions.

In the following, we denote by \bar{x}_2^* the value that maximizes the number of successful preambles' transmissions, that is, $\bar{x}_2^* = -1/\ln q_N$. Fig.2 shows the evolution of function f .

3.4 | Controller synthesis

Having described the system's dynamics, let's now study its controllability.

It can be clearly noted that the system, described in equations (4-7), is not controllable as a whole, as there is no means to act on the arrivals and, thus, on the number of the devices in the states $x_{1,i}(t)$ and $x_{1,L,i}(t)$, for all i .

In the following, we then propose to study the partial controllability³⁶ of the system. We consider the subsystem represented by equations (6) and (7), which corresponds to subsystem 2, as indicated in Fig.1. For the sake of simplification, we apply the same EAB factor P or P_{eab} for the different classes, i.e. $p_i = P$ for all $i \in \mathbb{C}$. Let, also, assume that $\sum_{i=1}^K x_{1,i} = x_1$.

For this subsystem's dynamics, we can write, in its state space,

$$\frac{dX}{dt} = A(X)X + B(X)P, \quad (18)$$

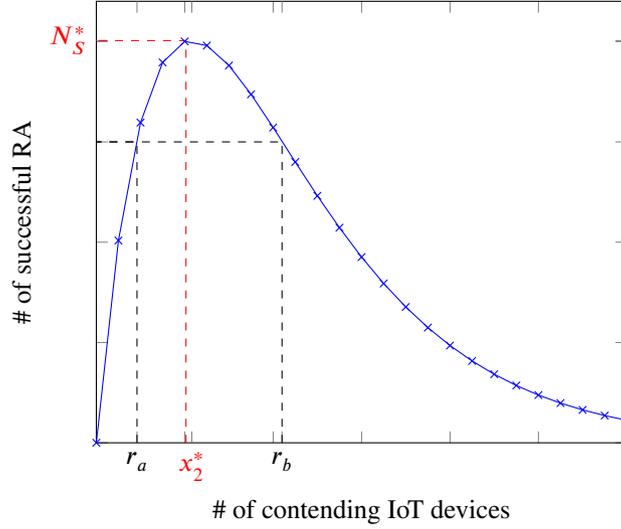


FIGURE 2 Successful RA vs number of contending devices. The analysis of the function $f(x) = xq^{x-1}$, for $x \geq 0$, where $0 < q < 1$, shows that it has an optimum N_S^* corresponding to $x_2^* = -1/\ln q_N$ number of devices. Apart from this optimum, it corresponds to a particular number of successful RA two possible solutions (r_a, r_b).

where

$$X = \begin{bmatrix} x_2 \\ x_{2,L} \end{bmatrix}, B = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$$

and

$$A = \begin{bmatrix} -(\theta + \theta_1) - q_N^{x_2-1} \ln(q_N)x_2 - q_N^{x_2-1} & \theta_2 \\ \theta_1 & -\theta_2 \end{bmatrix}.$$

Matrices A and B are, respectively, the system matrix and the control matrix. Matrix A relates to how the current state affects the state change. However, matrix B determines how the system input affects the state change. It can be easily verified that the subsystem described above is controllable (as the rank of $[B \ AB] = 2$). Consequently, the whole system is partially controllable in its state variables x_2 and $x_{2,L}$. If these states are observable, one can deduce that a feedback controller can be designed to make the variables, and in particular $x_2(t)$, converge to the optimal value, as defined in 3.3.

The subsystem described in equations (6-7) is non linear. Instead of proposing a non-linear control technique, as we did in a previous work²⁶, we propose, in what follows, the use of simple Proportional Integral Derivative (PID) controller³⁷, as we are focusing on the estimation of the state variables, in this paper, and not on its control.

Instead of using a continuous PID controller, as suggested by the model, we opted to use a discrete version. Indeed, the real system that we control works in discrete time (i.e. frame by frame processing) and since the PID controller is agnostic of the model, it is more appropriate to use a discrete version.

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

$$P_{\text{eab}}(n) = K_p e(n) + K_i \sum_{k=0}^n e(k) + K_d (e(n) - e(n-1)), \quad (19)$$

where n , P_{eab} , e , K_p , K_i and K_d represent respectively the instant, the controller output, the difference between the measured value – which will be described in the following section – and the targeted value (i.e. optimal value obtained in 3.3), the proportional gain, the integral gain and the derivative gain.

The main objective behind the PID controller is to make the total number of IoT devices attempting the access (i.e. x_2) converge to the optimal one x_2^* . The analysis of the impact of the controller on the estimation accuracy will be analyzed in the performance evaluation section.

4 | FINE-GRAINED ESTIMATION OF THE NUMBER OF IOT DEVICES

Optimizing network resources' usage necessarily requires the convergence of the number of devices attempting the access to the optimal number determined in the previous section. To reach such objective, one need to be able to effectively block the devices before they even attempt access. Barring them certainly implies to know the number of terminals wanting to access the network, those which are in the states variables x_1 or $x_{1,L}$. However, the devices in these states are not connected yet and thus invisible from the network's perspective.

We propose, in this paper, to estimate the number of IoTs, in state x_1 , indirectly via the number of terminals, in state x_2 , which successfully access the network. Specifically, we propose, from the number of successes, to estimate the terminals that attempted the access. This estimate will then be used to refine the estimate of the number of terminals wanting to access.

4.1 | Estimation of the number of contending devices (x_2)

Several methods in the literature proposed estimating the number of terminals that attempted the access, as introduced in the state of the art section. The originality of our approach lies in the use of the "balls into bins" model, which perfectly represents the access process. Thus, based on the model, defined in section 3.1, we can certainly infer a first estimator of the number of devices attempting the access (i.e. the number of terminals in state variable x_2). Indeed, the number of devices succeeding in the access (N_S), the number of unused preambles (N_I), and the number of collided preambles (N_F) can give an idea of the number x_2 .

Algorithm 1 Optimized x_2 estimation

```

1: global:  $N, N_S^*, N_S, N_F, N_I$ , and  $N_I^i, \forall i \in \mathbb{C}$ 
2:
3: function X2ESTIMATE( $n$ )
4:   require:  $\delta, \eta, \beta, N_{I,\min}, N_{I,\max}$ , samples
5:    $x_{2,\min}[n] \leftarrow N_S[n] + 2N_F[n]$ 
6:   if  $N_S[n] > N_S^*$  then
7:      $N_S[n] = N_S^*$ 
8:   end if
9:   if  $N_I[n] \neq 0$  then
10:     $x_{2,\text{idle}}[n] \leftarrow \frac{\ln(\frac{N_I[n]}{N})}{\ln q_N}$ 
11:     $\tilde{x}_2[n] \leftarrow \max(x_{2,\min}[n], x_{2,\text{idle}}[n])$ 
12:   else
13:     $x_{2,\text{success}}[n] \leftarrow \frac{W(q_N \ln q_N N_S[n])}{\ln q_N}$ 
14:     $\tilde{x}_2[n] \leftarrow \max(x_{2,\min}[n], x_{2,\text{success}}[n])$ 
15:   end if
16:    $\hat{x}_2 \leftarrow (1 - \eta)\hat{x}_2 + \eta\tilde{x}_2[n]$ 
17:    $\hat{x}_{2,i}[n] \leftarrow \frac{N_S^i[n]}{N_S[n]} \times \hat{x}_2[n], \forall i \in \mathbb{C}$ 
18:   return  $\hat{x}_2, \hat{x}_{2,i} \forall i \in \mathbb{C}$ 
19: end function

```

In order to avoid very large oscillations in the measurement of the parameters N_S , N_I and N_F , we propose in the following to consider one measurement per second, which corresponds to 100 frames. This ensures that the measures reflect reality and are therefore not the result of a completely random phenomenon.

Since a successful preamble concerns only one device and a collided preamble will concern at least two devices, a minimal number of devices $x_{2,\min}$ can be determined by $N_S[n] + 2N_F[n]$ (see line 5). This number represents a lower bound, which could be improved using the proposed model.

From equations (1) and (2), one can deduce x_2 knowing the other values, which can be measured by the access point. Since equation (1) presents two solutions, it is preferable to estimate this number from equation (2). Indeed, the solution of equation (2) is equal to $\ln(N_I/N)/\ln(q_N)$, which is solvable only when we have the number of N_I different from 0 (see line 10).

When there is no more preamble available ($N_I = 0$), reflecting a large number of IoT objects that attempted access, it is possible to measure the extent of congestion by looking at the number of terminals that have successfully accessed. It is in this sense that we consider only the stable solution of equation (1) (see Algorithm 1 for more details). The estimation of the number of contending devices \hat{x}_2 is, then, corrected using the factor expressed in line 16. The correction consists in considering the average (Exponential Weight Moving Average “EWMA”) values instead of the instantaneous one.

Note that the absence of unused preambles and the absence of successful IoT devices may lead to an underestimation of the number of terminals within the state variable x_2 . Indeed, in this case there is no way to really estimate the extent of the congestion. Therefore, it is necessary to couple this estimate with a relatively effective controller.

The estimation of the number of devices per class can, then, be obtained in line 17. We just assumed here that the proportion of successes for each class corresponds to the proportion of devices attempting the access.

4.2 | Crowd-based estimation of the number of IoT devices wanting the access (x_1)

To the best of our knowledge, the estimate of the number of equipment wanting to connect has not been addressed in the literature. The difficulty of this estimation is the absence of direct parameters to estimate such number. However, the blocking effects (i.e. barring factor) combined with the number of successful accesses certainly give an indication, more or less precise, of the number of IoT devices that have attempted the access. In fact, when the blocking factor is not too big (for very small values of P_{eab}), the number of devices wanting to connect can be directly derived from the number of devices who just accessed the x_2 state, since it represents a known fraction P_{eab} of the total number of devices wanting to access.

The number of devices who just reached the state variable x_2 can be approximated by the difference between the number of estimated devices $\hat{x}_2[n]$ and the IoT devices that was already there, which might be represented by the difference between the last estimate value $\hat{x}_2[n-1]$ and those that succeeded in the access ($N_S[n-1]$) (see line 9 of Algorithm 2). The number of devices wanting to connect for each class can be deduced in a similar way (see line 11). Note that we are not considering here the number of devices aborting the transmission as a good system should completely avoid such case.

When the barring factor is too big (the access probability too small) the estimation of the number of devices cannot be estimated using the methodology explained above. Therefore, we propose using crowd sourcing to estimate such number. In fact, as there is no direct or indirect ways to estimate such number, we propose to overload the number of connections and accesses attempts in the connection request message sent by each IoT device. In this way, the reception of the connection requests’ messages by the eNodeB becomes very informative since it allows to know the average number of attempts at each step, and thus the congestion level of the network.

We start by defining an ideal realistic target for the average number of attempts N_{eab}^* , which should be more than one and smaller than three to have reasonable number of attempts. We, then, defined two thresholds: a minimum threshold N_{min} and the maximal threshold N_{max} around which the average should oscillate. An average number of attempts bigger than N_{max} means an underestimation of the congestion level, and requires, then, an adjustment upwards of the number of devices by adding to it a small quantity $\beta\hat{x}_1[n]$. On the other hand, an average number small than N_{min} means an overestimation of the congestion level, which requires the adjustment of the number of devices downwards. The same technique was used to estimate the number of devices of the different classes.

In the current standard, the number of connections and accesses attempts may be overloaded into the “RRC connection request”, which is sent just after a successful preamble transmission (i.e. reception of the “Random access response” (RAR) from the eNodeB) through the CCCH logical channel³⁸. To be compliant with the actual version of the standard, 8 bits could be taken from the 40 bits “randomValue” field of the request. 3 bits could be reserved for each number (this gives the possibility to record up to 7 access attempts, which is more than enough for a stable system.) and 2 bits (set to 0) to indicate that the last 6 bits will be used to store such numbers.

Algorithm 2 Optimized x_1 estimation

```

1: global:  $N_{eab}^*$ ,  $\hat{x}_2$ ,  $N_S$ ,  $N_S^*$ ,  $P_{acb}$ , and  $N_S^i$ ,  $\hat{x}_2^i$ ,  $N_{acb}^i$ ,  $\forall i \in \mathbb{C}$ 
2:
3: function X1ESTIMATE( $n$ )
4:   require:  $\alpha_1 \in [0, 1]$ ,  $\alpha_2 > 1$ ,  $\eta$ ,  $\epsilon = 10^{-6}$ ,  $\beta$ 
5:    $N_{min} \leftarrow \alpha_1 N_{acb}^*$ 
6:    $N_{max} \leftarrow \alpha_2 N_{acb}^*$ 
7:    $\bar{x}_1 \leftarrow (1 - \eta)\bar{x}_1 + \eta\hat{x}_1[n - 1]$ 
8:    $\bar{x}_1^i \leftarrow (1 - \eta)\bar{x}_1^i + \eta\hat{x}_1^i[n - 1]$ ,  $\forall i \in \mathbb{C}$ 
9:   if  $P_{eab}[n] \geq \epsilon$  then
10:      $\hat{x}_1[n] \leftarrow \frac{\hat{x}_2[n] - (\hat{x}_2[n - 1] - N_S[n - 1])}{P_{acb}[n]}$ 
11:      $\hat{x}_{1,i}[n] \leftarrow \frac{\hat{x}_{2,i}[n] - (\hat{x}_{2,i}[n - 1] - N_S^i[n - 1])}{P_{acb}[n]}$ ,  $\forall i \in \mathbb{C}$ 
12:   else
13:     if  $N_{acb} \geq N_{max}$  then
14:       return  $\hat{x}_1[n] \leftarrow \bar{x}_1 - \beta\hat{x}_1[n - 1]$ 
15:     else
16:       if  $N_{acb} \leq N_{min}$  then
17:         return  $\hat{x}_1[n] \leftarrow \bar{x}_1 + \beta\hat{x}_1[n - 1]$ 
18:       else
19:         return  $\hat{x}_1[n] \leftarrow \bar{x}_1$ 
20:       end if
21:     end if
22:     if  $N_{acb}^i \geq N_{max}$  then
23:       return  $\hat{x}_1^i[n] \leftarrow \bar{x}_1^i - \beta\hat{x}_1^i[n - 1]$ ,  $\forall i \in \mathbb{C}$ 
24:     else
25:       if  $N_{acb}^i \leq N_{min}$  then
26:         return  $\hat{x}_1^i[n] \leftarrow \bar{x}_1^i + \beta\hat{x}_1^i[n - 1]$ ,  $\forall i \in \mathbb{C}$ 
27:       else
28:         return  $\hat{x}_1^i[n] \leftarrow \bar{x}_1^i$ ,  $\forall i \in \mathbb{C}$ 
29:       end if
30:     end if
31:   end if
32: end function

```

5 | PERFORMANCES EVALUATION

In order to evaluate the accuracy of the estimations, we built from scratch a discrete events' simulator in C representing the access of multi-class IoT devices in a Telco network. Table 1 summarizes the parameters used under this simulator.

We assume that there is one access point and that IoT devices are activated according to a Poisson traffic model where inter-arrivals are exponentially distributed. We considered two classes of service: the first class refers to emergency and prioritized applications and the second class refers to non prioritized applications (e.g. smart metering applications) which are delay tolerant. We also adopt a RACH configuration where one RACH opportunity occurs every 10ms with 54 preambles at each opportunity. When a RACH trial is declared unsuccessful, a device can retry the RA after a backoff time chosen uniformly between 0 and the backoff parameter fixed in Table 1. As described in the standard, a limited number of attempts (10 attempts) is considered here before aborting the transmission. Note that there is no dropout in the first phase and that the number of attempts is therefore unlimited. On the other hand, any failure in the EAB check blocks the device from a new trial during an ac-BarringTime.

In the following, we performed a series of experiments. In the first experiment, we evaluated the estimation accuracy of the number of contending devices as a function of the time. In the second set of experiments, we analyzed the accuracy of the

estimation for various loads. The obtained average values and the confidence intervals were computed for 30 experiments for each network's load value. In the third set of experiments, we test the impact of our proposed controller on the estimation's method.

Note that the error bars (in subsections 5.2 and 5.3) are obtained for 95% confidence intervals.

TABLE 1 Basic simulation parameters

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

5.1 | Estimation efficiency

Fig.3 and Fig.4 show respectively the existing gap between the real number x_1 and the estimated one \hat{x}_1 , and between the real value x_2 and the estimated one \hat{x}_2 . It can be seen that only very small fluctuations exist between x_1 and \hat{x}_1 and between x_2 and \hat{x}_2 , which proves the accuracy of our estimation.

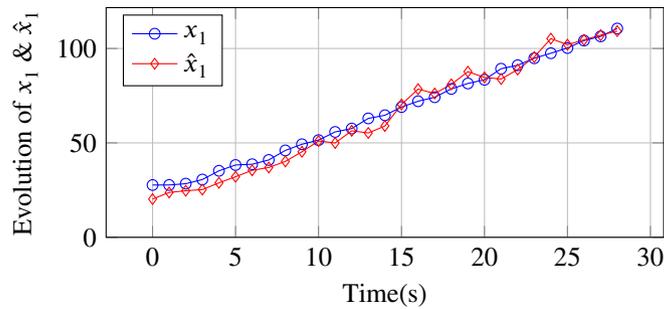


FIGURE 3 Evolution of x_1 & \hat{x}_1 as a function of time

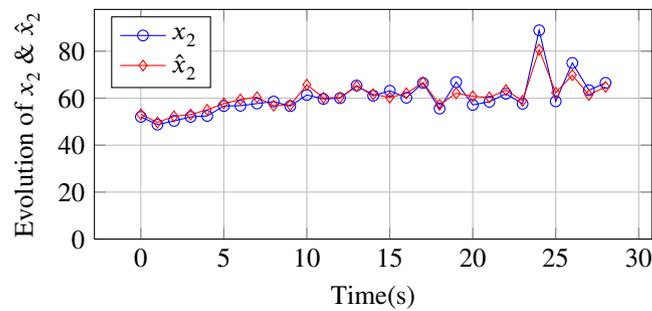


FIGURE 4 Evolution of x_2 & \hat{x}_2 as a function of time

Fig.5 depicts the evolution of the estimated \hat{x}_1 regarding the sum of the estimated $\hat{x}_{1,i}, \forall i \in \mathbb{C}$. We can easily remark that \hat{x}_1 is nearly the same as the sum of $\hat{x}_{1,i}$ estimated separately per class. This is one of the major strength of our crowd-based estimation's method.

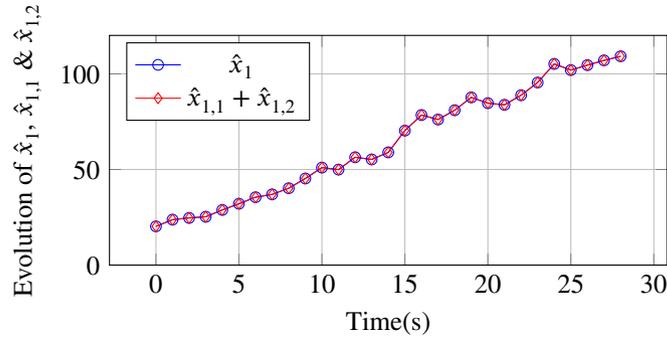


FIGURE 5 Evolution of \hat{x}_1 , $\hat{x}_{1,1}$ & $\hat{x}_{1,2}$ as a function of time

Similarly, as shown in Fig.6, the estimated \hat{x}_2 , corresponds to the sum of $\hat{x}_{2,i}$ estimated separately per class. This is a direct consequence of our assumption that the proportion of successes for each class corresponds to the proportion of devices attempting the access.

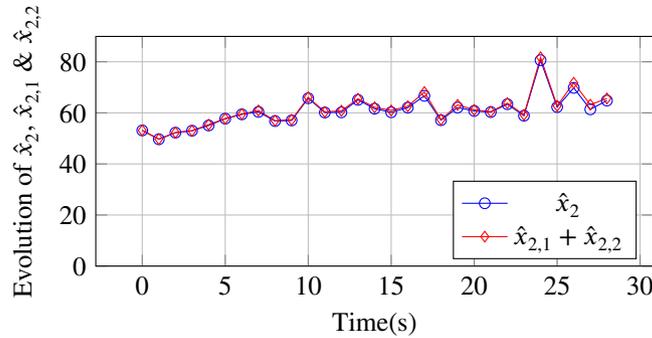


FIGURE 6 Evolution of \hat{x}_2 , $\hat{x}_{2,1}$ & $\hat{x}_{2,2}$ as a function of time

5.2 | Estimation accuracy for various loads

Fig.7 depicts the evolution of the estimation error of x_1 , $x_{1,1}$ and $x_{1,2}$ for various loads. It can be clearly seen that when the network's load increases, the average estimation error (of x_1 , $x_{1,1}$ and $x_{1,2}$) increases. More specifically, a remarkable increase ($\simeq 13\%$) is noted when $\lambda = 20$ and then we obtain a more accurate estimation ($\simeq 5\%$). In fact, when the network is more congested, x_1 (and respectively $x_{1,1}$ and $x_{1,2}$) tends to reach stable values and then the estimation's error is reduced. We also observe that the estimation error of $x_{1,1}$ is very slightly higher than x_1 and $x_{1,2}$, because of the reduced arrival rate of such emergency applications. Indeed, when arrivals are infrequent, there are less observed data, the confidence level is reduced and then the estimation's error is increased.

Fig.8 shows the evolution of the estimation error of x_2 , $x_{2,1}$ and $x_{2,2}$ for various loads. We can easily remark that when the network's load increases, the average estimation error (of x_2 , $x_{2,1}$ and $x_{2,2}$) increases slightly, but doesn't exceed 8%. Besides, the estimation error of $x_{2,1}$ is slightly higher than x_2 and $x_{2,2}$, because of the infrequent arrivals of such class. The same explanation as above justifies this behaviour.

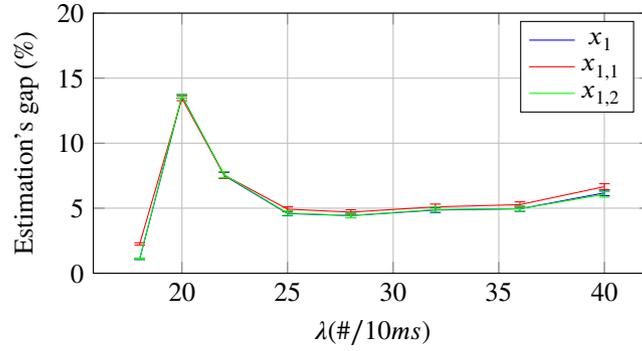


FIGURE 7 Evolution of the estimation error of x_1 vs λ

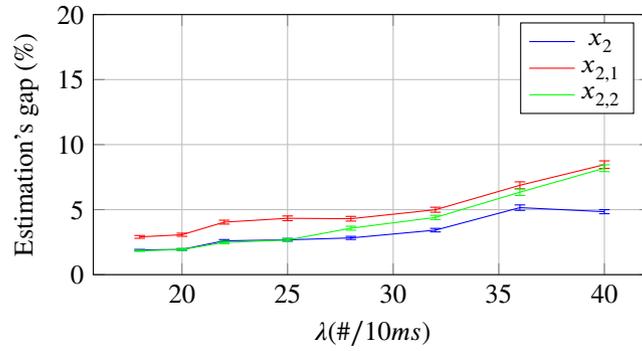


FIGURE 8 Evolution of the estimation error of x_2 vs λ

5.3 | Impact of the controller on the estimation accuracy

To show the impact of using the proposed PID adaptation controller on our estimation's method, we compare, in the following, its behaviour with simple approaches using fixed EAB probabilities.

Fig.9 and Fig.10 show respectively the evolution of the estimation error of x_2 and x_1 for various loads. We can easily check the rapid degradation of the estimation's accuracy of x_2 ($\simeq 70\%$ of estimation's error) and x_1 ($\simeq 100\%$ of estimation's error) when a fixed EAB approach is applied and when the network's load increases. Only 5% of estimation's error is measured when an adaptive approach is used, independently from the network's load. To justify such behaviour, we show in Fig.11 and Fig.12, respectively, the evolution of the number of Idle preambles N_I and successful preambles N_S for various loads.

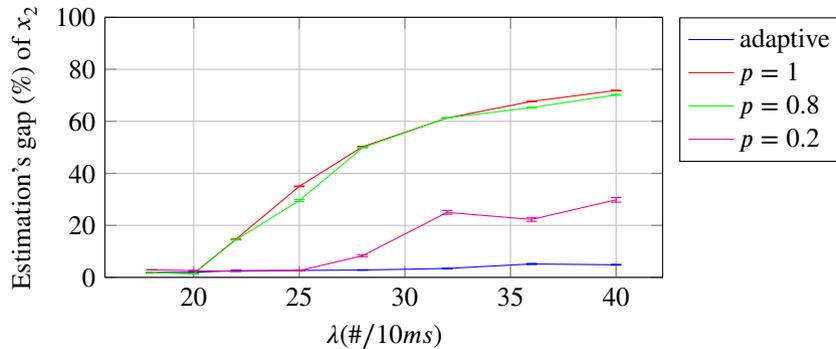


FIGURE 9 Evolution of the estimation error of x_2 vs λ

We can clearly remark that when the network is undergoing a severe congestion (i.e our system is no more stable) and if a simple EAB approach is adopted, N_I is nearly equal to zero (i.e. resources' over-utilization) and N_S decreases gradually

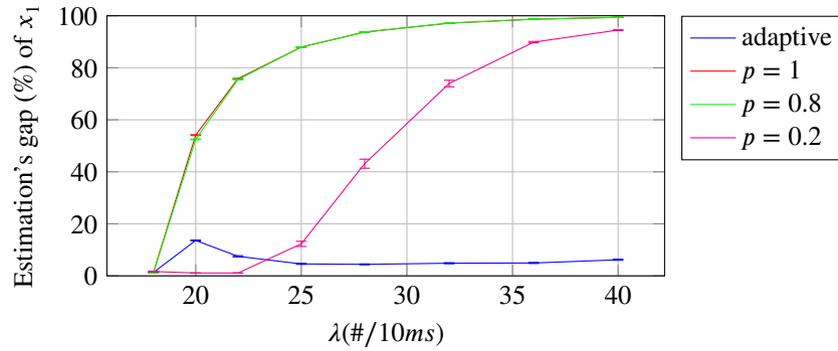


FIGURE 10 Evolution of the estimation error of x_1 vs λ

and becomes intolerable. However, when an adaptive PID controller is applied, N_I and N_S are slightly reduced but tend to reach stable values and then a more stable system, even if the network's load increases. In such a case, our estimation's method remains efficient. Nonetheless, this efficiency is deteriorated when a fixed EAB approach is used. Hence, the efficiency of our estimation's solution is closely related to the action of the controller in insuring the stability of the system.

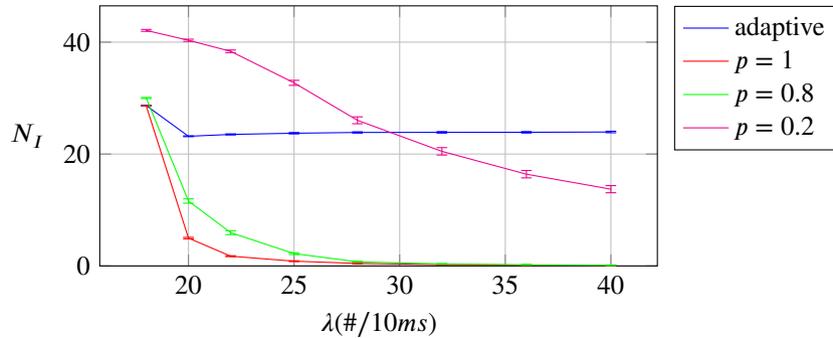


FIGURE 11 N_I vs λ

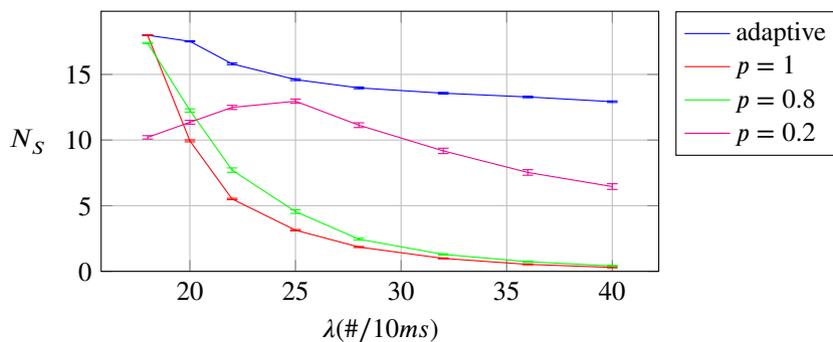


FIGURE 12 N_S vs λ

6 | CONCLUSION

One of the key challenges associated with the growth of connected IoT devices in 5G networks is to deal with the congestion collapse issue, particularly when a massive amount of simultaneous IoT arrivals occurs. To limit such a burden, many current

solutions consider using barring probabilities at the access to reduce the amount of incoming devices. Nonetheless, the efficiency of such techniques remain closely related to an accurate knowledge of the number of contending IoT devices. Hence, a better estimation of the number of devices willing to connect, would efficiently help to overcome the risk of congestion.

Based on a model of access for multi-class IoT devices, we have, in this paper, proposed to enrich the connection requests with the number of access attempts to make the base station aware of the congestion's status. Using such information, we propose a novel crowd sourcing recursive algorithm to estimate the number of devices.

The obtained results demonstrated the efficiency of our proposition as we obtained estimated values very near to the real ones. We also investigated the trade-off existing between the estimation's accuracy and the use of an efficient controller. Indeed, the efficiency of the proposed method of estimation requires a stable system, and thus an efficient controller should be activated.

References

1. G.Y. Lin, and H.Y. Wei, *Flexible 5G M2M network access with cognitive RAN: Survey and design principles*, IEEE Conference on Standards for Communications and Networking (CSCN), October 2015.
2. C.S. Bontu, S. Periyalar, and M. Pecun, *Wireless Wide-Area Networks for Internet of Things: An Air Interface Protocol for IoT and a Simultaneous Access Channel for Uplink IoT Communication*, IEEE Vehicular Technology Magazine, March 2014.
3. S.Y. Lien, K.C. Chen, and Y. Lin, *Toward Ubiquitous Massive Accesses in 3GPP Machine-to-Machine Communications*, IEEE Communications Magazine, April 2011.
4. Z. Abbas, and W. Yoon, *A Survey on Energy Conserving Mechanisms for the Internet of Things: Wireless Networking Aspects*, Sensors journal, 2015.
5. U. Raza, P. Kulkarni, and M. Sooriyabandara, *Low Power Wide Area Networks: An Overview*, IEEE Communications Surveys & Tutorials, Vol. 19, Issue 2, Second-quarter 2017.
6. L. Zhang, A. Ijaz, P. Xiao, and R. Tafazolli, *Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km² Area*, IEEE 85th Vehicular Technology Conference (VTC Spring), June 2017.
7. F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. MeliaSegui, and T. Watteyne, *Understanding the Limits of LoRaWAN*, IEEE Communications Magazine, January 2017.
8. R. Boisguene, S.C. Tseng, C.W. Huang, and P. Lin, *A survey on NB-IoT downlink scheduling: Issues and potential solutions*, 13th Wireless Communications and Mobile Computing Conference (IWCMC), June 2017.
9. J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, *Narrowband Internet of Things: Implementations and Applications*, IEEE Internet of Things Journal, Vol. 4, Issue 6, December 2017.
10. Y.D. Beyene, R. Jantti, O. Tirkkonen, K. Ruttik, S. Iraji, A. Larmo, T. Tirronen, and J. Torsner, *NB-IoT Technology Overview and Experience from Cloud-RAN Implementation*, IEEE Wireless Communications, Vol. 24, Issue 3, June 2017.
11. *3GPP TR 37.868, RAN Improvements for Machine-type Communications, V.11.0.0, March 2013.*
12. *3GPP TR 37.868 V11.0.0*, Technical Specification Group Radio Access Network; Study on RAN Improvements for Machine-type Communications (Release 11).
13. A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb *Cellular-based machine-to-machine: overload control*, IEEE Network, Vol. 26, no. 6, pp. 54-60, November 2012.
14. O. Arouk, A. Ksentini, and T. Taleb, *Group Paging-based Energy Saving for Massive MTC Accesses in LTE and Beyond Networks*, IEEE Journal on Selected Areas in Communications, 2016.
15. Z. Wang and V. W. S. Wong, *Optimal Access Class Barring for Stationary Machine Type Communication Devices With Timing Advance Information*, in IEEE Transactions on Wireless Communications, Vol. 14, No. 10, pp. 5374-5387, October 2015.

16. M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, G. Rubino, and S. Tabbane, *Dynamic adaptive access barring scheme for heavily congested M2M networks*, 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM' 16), Malta, November 2016.
17. M.Z. Hasan, H. Al-Rizzo, and F.M. Al-Turjman, *A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Multimedia Wireless Sensor Networks*, IEEE Communications Surveys and Tutorials, 2017.
18. F.M. Al-Turjman, *Cognitive Routing Protocol for Disaster-inspired Internet of Things*, Elsevier Future Generation Computer Systems, 2017.
19. M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, and S. Tabbane, *On the risk of congestion collapse in heavily congested M2M networks*, the International Symposium on Networks, Computers and Communications (ISNCC' 16), Hammamet, Tunisia, May 2016.
20. F. Hussain, A. Anpalagan, and R. Vannithamby, *Medium access control techniques in M2M communication: survey and critical review*, Wiley Transactions on Emerging Telecommunications Technologies, 2014.
21. ETSI TS 122 01, V11.3.0, *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Service accessibility (3GPP TS 22.011 version 11.3.0 Release 11)*, April 2013.
22. *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.
23. M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, *Massive Non-Orthogonal Multiple Access for Cellular IoT: Potentials and Limitations*, IEEE Communications Magazine, September 2017.
24. A. Li, Y. Lan, X. Chen, and H. Jiang, *Non-Orthogonal Multiple Access (NOMA) for Future Downlink Radio Access of 5G*, China Communications, vol. 12, no. Supplement, pp. 28-37, December 2015.
25. M. Shirvanimoghaddam, M. Condoluci, M. Dohler, and S. J. Johnson, *On the Fundamental Limits of Random Non-orthogonal Multiple Access in Cellular Massive IoT*, IEEE Journal on Selected Areas in Communications, 2017.
26. M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, G. Rubino, and S. Tabbane, *Applying nonlinear optimal control strategy for the access management of MTC devices*, the 13th Annual IEEE Consumer Communications & Networking Conference (CCNC' 16), Las Vegas, USA, January 2016.
27. A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, *Cellular-based machine-to-machine: overload control*, IEEE Network, Vol. 26, no. 6, pp.54-60, November 2012.
28. H. He, Q. Du, H. Song, W. Li, Y. Wang, and P. Ren, *Traffic-aware ACB scheme for massive access in machine-to-machine networks*, IEEE ICC, 2015.
29. H. He, P. Ren, Q. Du, L. Sun, and Y. Wang, *Traffic-aware overload control scheme in 5G ultra-dense M2M networks*, Transactions on emerging telecommunications technologies, Vol. 28, Issue 9, September 2017.
30. Y. Pang, G. Lin, and H. Wei, *Context-aware Dynamic Resource Allocation for Cellular M2M Communications*, Internet of things journal, 2015.
31. G. C. Madueno, N. K. Pratas, C. Stefanovic, and P. Popovski, *Massive M2M Access with Reliability Guarantees in LTE Systems*, IEEE ICC, 2015.
32. M. Bouzouita, Y. Hadjadj Aoul, N. Zangar, and S. Tabbane, *Leveraging crowd sourcing technique for an optimized M2M access during emergency situations*, ICT-DM, Vienna, Austria, December 2016.
33. M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, G. Rubino, and S. Tabbane, *Multiple Access Class Barring factors Algorithm for M2M communications in LTE-Advanced Networks*, ACM/MSWIM, 2015.
34. R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, *On the Lambert W function*, Advances in Computational Mathematics, Vol. 5, Issue 1, pp 329-359, December 1996.

-
35. wikipedia, Newton's method, https://en.wikipedia.org/wiki/Newton's_method.
 36. A. E. BASHIROV, N. MAHMUDOV, N. SEMI, and H. ETIKAN, *Partial controllability concepts*, International Journal of Control, Vol. 80, No. 1, January 2007.
 37. K.J. Astrom, and T. Hagglund, *Advanced PID Control*, ISA-The Instrumentation, Systems, and Automation Society, 2006.
 38. ETSI TS 136 331, V14.2.2, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 14.2.2 Release 14)*, May 2017.

