Adaptive load control for IoT based on satellite communications

Hugo Chelle, Michael Crosnier, Riadh Dhaou, André-Luc Beylot

Adaptive load control for IoT based on satellite communications

Hugo Chelle*‡, Michael Crosnier*, Riadh Dhaou‡ and André-Luc Beylot‡
*Airbus Defence and Space, Telecom System Department, France
‡Université de Toulouse, IRIT Lab, France
{hugo.chelle, michael.crosnier}@airbus.com, {riadh.dhaou, andre-luc.beylot}@irit.fr

***Abstract*— The Internet Of Things (IoT) market is growing more and more every year. Today, the number of IoT devices is estimated around 8 billion but forecasts announce 20 billion devices for 2020. Terrestrial or satellite communications systems are already deployed to answer the connectivity need. These systems rely on a Random Access CHannel (RACH) used either to send resource allocation requests or directly the useful message.**

Because of the number of IoT devices, the overload on the RACH is an emerging issue since it may cause a service outage. This is especially the case for IoT satellite systems because of the wide area covered by a single satellite. The Access Class Barring (ACB) is the load control mechanism used within the Narrow Band IoT. Unfortunately, no method was specified to compute the load control parameters.

In this paper, in the context of a satellite IoT system, we propose a method to compute dynamically ACB based load control parameters. Thanks to our method, the load control mechanism reach excellent results regarding transmission reliability and energy consumption for various traffic scenarios.

I. Introduction

Nowadays, we notice an expansion of the Internet of Things (IoT) market. Billions of devices (around 8 billion [1]) are already deployed. According to forecasts, the number of devices should still grow in the coming years to reach up to 20 billion in 2020 [2].

Several terrestrial cellular IoT systems are already operational based for instance on Sigfox or LoRa radio technology. The Third Generation Partnership (3GPP), has developed the Narrow Band IoT (Nb-IoT) [3], a new cellular air interface dedicated to IoT communications. IoT access providers also extend in space thanks to satellite constellations (Orbcomm, Iridium) or Geostationary satellites. Satellites are essential for IoT market to reach its full potential. Satellites ensure a worldwide connectivity which is essential for many applications such as freight tracking or applications based on sensors scattered in remote areas. All this creates a very dynamic ecosystem attractive to research as there are still many challenges to overcome.

Communications systems rely on a Random Access CHannel (RACH) which is used either to send resource allocation requests (as in Nb-IoT [3] and the Demand Assigned Multiple Access in DVB-RCS2 [5]) or to transmit the useful message directly (pure random access protocol as Sigfox and LoRa). In the IoT context, because of the colossal number of devices expected in the coming years, the overload on the RACH appears as a major challenge to overcome since it may cause a service outage. To counteract the RACH overload, one may oversize the RACH capacity or use load control algorithms. The Access Class Barring (ACB) [6] was chosen by the 3GPP to counteract overload in the NB-IoT [3]. The ACB mechanism is composed of a RACH access probability and a barring time for the devices blocked by the load control. These load control parameters are broadcasted by the base station to reduce the RACH load.

In this paper, we study load control mechanisms for a satellite system dedicated to the IoT. Given the satellite coverage, load control mechanisms are essential for satellites since millions of devices could be within their coverage (50 000 devices should be deployed within a NB-IoT cell [3]). We consider a system with a time slotted RACH and a downlink to carry signaling such as acknowledgments and load control parameters. Moreover, we consider ACB like load control parameters which are updated at each time slot. We use a time slotted random access since it is commonly the case by terrestrial or satellite systems (Nb-IoT [3], DVB-RCS2 [5]). We have chosen to focus on the ACB since it is the most widely used (LTE, NB-IoT). Note that it is possible to update load control parameters at each time slot, in Nb-IoT, System Information Block containing ACB parameters is broadcasted for each RACH time slot [3]. This operation does not influence devices batteries since devices read this information only to send a message on the RACH.

The 3GPP did not specify a method to compute ACB load control parameters. We propose, in this paper, a mechanism to dynamically adapt ACB-based load control parameters according to the number of messages sent to the satellite by time slot on the RACH (also called the RACH load). The main contributions introduced by the paper are:

- An algorithm to estimate the RACH load of future time slots. The novelty is the distinction between the new devices, the devices in retransmission and the devices previously blocked by the load control. Thanks to this approach, the algorithm estimates accurately the future RACH load.
- A method to calculate the access probability which optimizes the number of messages correctly received by satellite (i.e. the throughput) during overloaded time slots.
- A method to compute the barring time parameters enabling to save up devices energy compared to the literature.

This paper is composed as follows: Section II describes the state of the art of load control mechanism, Section III sketches the communication system model, Section IV presents our load control mechanism. Then, Section V highlights the mechanism performance, and in section VI we summarize our contributions.

II. Related work

We focus on the related work on ACB [6] load control mechanisms. An ACB mechanism is composed of two load control parameters: the probability of access p_{ACB} and the barring time t_{ACB}. A device is allowed to transmit if a random number drawn uniformly in the interval $[0, 1]$ is less than or equal to the probability of access p_{ACB}, this step is called the access test. Devices failing the access test will retry to transmit t_{delay} later, $t_{delay} = (0.7 + 0.6X)\, t_{ACB}$, where X is a random number uniformly distributed within the interval $[0, 1]$.

Previous papers consider terrestrial communications systems such LTE-Advanced, LTE-Machine or Nb-IoT. The slotted Aloha is used for the RACH in these systems. So, to the best of our knowledge, no study was carried out to examine the impact, of a new access method, on the ACB load control mechanism. We may divide ACB-based mechanisms into two types: static or dynamic. With static ACB, the same load control parameters are applied during overloaded time slots whereas, for dynamic ACB, the load control parameters are adapted to the overload.

The authors of [7] and [8] highlight static ACB performance for a unique traffic scenario defined by the 3GPP in [9]. The authors have shown that proper performance could be obtained if the static load control parameters are selected wisely. The paper [10] proposes an analytical framework to evaluate the performance of static ACB. This paper shows that, for given ACB parameters, the transmission reliability drops when the number of devices performing the random access procedure rises. Therefore, the selection of static ACB parameters depends on the overload traffic profile. Static ACB load control mechanism should be used only when a single traffic scenario occurs on a base station/satellite.

Thus, dynamic ACB load control is studied to accommodate to overloads. In papers [11] and [12], the authors detail algorithms where the probability of access (p_{ACB}) is computed dynamically thanks to an estimation of the number of backlogged devices. These algorithms allow the load control mechanism to adapt efficiently to overload variations, but we notice two drawbacks. In [12], the algorithms must know the total number of devices activated during the overload. In practice, this information is unknown for base stations or satellites. The algorithms of both papers [11] [12] do not consider barring time parameter t_{ACB}. Thus, despite overload, devices continue to try to transmit on the RACH at each time slot which impacts devices batteries. Indeed, devices stay awake for many time slots and persist in trying to transmit on the RACH. This is a waste of energy; devices should wait until the end of the overload before to retry to transmit on the RACH. In IoT, the device battery has to last several years thanks to a basic AA battery. Hence, the load control mechanism shall not impact the device energy consumption.

Therefore, in this paper, we detail the computation of the optimal access probability for a typical satellite random access method. Furthermore, we consider the computation of the barring time parameters to avoid energy waste.

III. Communication system modeling

Let us consider, a geostationary satellite system dedicated to IoT communication. We focus on the modeling of the RACH (of the return link) which is described in the following. Devices can send on this RACH either resource allocation requests or the useful packet.

A. Access method

We choose the Spread Spectrum - Multi Frequency - Contention Resolution Diversity Slotted Aloha (SS-MF-CRDSA) [13] access method for the random access channel. The SS-MF-CRDSA is an improvement of the CRDSA [14] regarding throughput, power, and energy efficiency useful to meet IoT requirements. We have chosen the SS-MF-CRDSA since the CRDSA is used for the random access channel of the DVB-RCS2 [5].

With the SS-MF-CRDSA, devices transmit $N_{rep} = 2$ replicas within a time slot; the uplink time slot is divided into $N_T = N_{rep}$ time sub-slots, each time sub-slot is divided into $N_F = 20$ frequency slots, the spreading factor is defined by $S_F = 2$. Devices transmit their packets of $b = 100$ bits with a QPSK modulation using a 3GPP FEC (rate $r = 1/3$), the received packets power follows a uniform distribution between $\left(\frac{E_b}{N0}\right)_{min} = 2$ dB and $\left(\frac{E_b}{N0}\right)_{max} = 12$ dB. The performance of the random access is given by [13].

B. MAC layer modeling

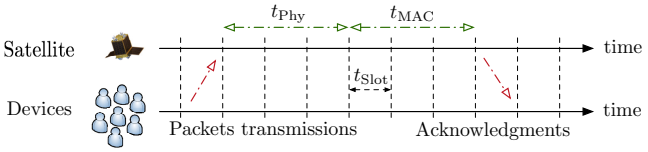

Fig. 1: Communications between the devices and the satellite, t_{Slot} is the slot duration

Figure 1 highlights the MAC layer modeling. We consider that :

- Devices are either in energy-saving mode (i.e. asleep) or in transmission mode.
- Devices are synchronized with the satellite after their awakening.
- Devices transmit one packet at a time since IoT devices have a very limited throughput requirement [3].

Packets transmitted during a time slot (by several devices) are processed by the satellite. Satellite processing is composed of the physical layer (duration t_{Phy}) and the MAC layer processing (duration t_{MAC}). At the satellite processing end, the satellite broadcasts acknowledgments. We suppose that

the satellite processing duration is constant. Indeed, from a hardware point of view, it is usually easier to always realize the same processing operations whatever the RACH load which leads to constant processing duration. Thus, devices know the time slot when they should receive their acknowledgment. If no acknowledgment is received, devices try to send again. We consider that devices have $N_{\text{Max Transmission}}$ attempts to transmit a message, above this limit, devices discard the message.

C. Load Control modeling

1) Satellite side: Right after the physical layer processing end, the satellite computes the new load control parameters. The duration of this computation is t_{LC}. Then, these new parameters are broadcasted to the devices.

2) Device side: Before the transmission, devices listen to the satellite to ensure the synchronization and to receive the newest load control parameters. Devices which succeed the access test mentioned during the introduction are allowed to transmit on the RACH. Whereas, the other devices will retry to transmit during a later time slot determined by the barring time parameters. Since the load control mechanism should not waste devices' energy, we consider that devices can be blocked by the load control at most $N_{\text{Max Block}}$. Above this limit, devices discard the packet to avoid an overconsumption of energy.

IV. Load control mechanism

A. Load control mechanism architecture

We propose to divide the load control architecture into several parts:

- The RACH load estimator. Its role is to estimate the number of packets transmitted on the RACH per time slot. We note $G(k)$ the RACH load during time slot k and $\widehat{G}(k)$ its estimation.
- The future prediction. Its objective is to estimate the RACH load during future time slots from current RACH load estimations.We note $G_T(k)$ the expected load during time slot k and $\widehat{G_T}(k)$ its estimation.
- The load control calculator. Its role is to compute new load control parameters from the expected load.

We consider two independent barring time parameters, the minimal barring time t_{min} and the spreading barring time t_{spread}. Hence, the transmission delay for devices blocked by the load control is $t_{delay} = t_{\text{min}} + Xt_{\text{spread}}$ where X is a random number drawn uniformly from $[0, 1]$. Note that in regular ACB,t_{min} and t_{spread} are linked, $t_{\text{min}} = 0.7t_{ACB}$ and $t_{\text{spread}} = 0.6t_{ACB}$. We have kept uncorrelated t_{min} and t_{spread} to better understand their respective impact. In the following, we note $p(k)$ (p is the access probability), $t_{\text{min}}(k)$ and $t_{\text{spread}}(k)$ the load control parameters applied during time slot k.

B. Load Estimator

We consider a generic RACH load estimator since realistic load estimators depend on the physical layer algorithms at the receiver side. Equation (1) highlights the load estimator considered during the simulations where $\beta \in [0, 1]$ (when $\beta = 0$ the load estimator is perfect and $\beta = 1$ leads to the maximal range of load estimations errors), $\mathscr{U}$ is the discrete uniform distribution, and $G(k)$ is the true RACH load during the time slot number k.

$$\widehat{G}(k) = \mathscr{U}\left(G(k) - \lfloor \beta G(k) \rfloor, G(k) + \lfloor \beta G(k) \rfloor\right), \ \beta \in [0, 1] \quad (1)$$

Thanks to this estimator, we can observe the impact of inaccurate RACH load estimations on the load control mechanism performance.

C. Estimation of the expected load

Description	Time	Number of Slots
Physical layer processing	t_{Phy}	$k_{\text{Phy}} = \lceil \frac{t_{\text{Phy}}}{t_{\text{Slot}}} \rceil$
MAC layer processing	t_{MAC}	$k_{\text{MAC}} = \lceil \frac{t_{\text{MAC}}}{t_{\text{Slot}}} \rceil$
Load control processing	t_{LC}	$k_{\text{LC}} = \lceil \frac{t_{\text{LC}}}{t_{\text{Slot}}} \rceil$
Minimal delay	t_{min}	$k_{\text{min}} = \lceil \frac{t_{\text{min}}}{t_{\text{Slot}}} \rceil$
Delay to spread devices	t_{spread}	$k_{\text{spread}} = \lceil \frac{t_{\text{spread}}}{t_{\text{Slot}}} \rceil$

TABLE I: Notations.

Table I summarizes important notations for the following. Suppose that the load control processing starts at time slot k. The load control objective is to compute the load control parameters applied during the future time slot k_{Estim}. To do so, the mechanism needs to estimate the expected load during the time slot k_{Estim}. The expected load estimation in a future time slot is done from RACH estimations $\widehat{G}$ of which the most recent is obtained for time slot k_{Last} because of the satellite onboard processing duration. Figure 2 highlights the index explanation.

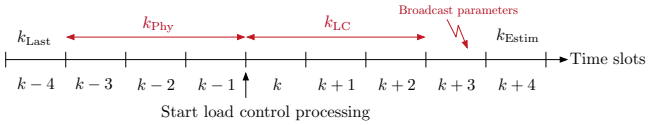

Fig. 2: Index explanation example, $k_{\text{Phy}} = 3$ and $k_{\text{LC}} = 3$.

We may break down $\widehat{G_T}(k_{\text{Estim}})$ as given by equation (2), where $G_N(l)$ is the number of new packets which are transmitted during time slot l, $G_R(l)$, is the number of packets that failed their transmission during previous time slots which are retransmitted during time slot l, and $G_B(l)$ is the number of packets previously blocked by the load control which are transmitted during time slot l.

$$\widehat{G_T}(k_{\text{Estim}}) = \widehat{G_N}(k_{\text{Estim}}) + \widehat{G_R}(k_{\text{Estim}}) + \widehat{G_B}(k_{\text{Estim}}). \quad (2)$$

It turns out that the method to estimate $G_R(k_{\text{Estim}})$ and $G_B(k_{\text{Estim}})$ is deterministic since the load control mechanisms knows $\widehat{G}(j)$, $p(j)$, $k_{\text{min}}(j)$, $k_{\text{spread}}(j)$ and $S(j)$ for $j \leq k_{Last}$. We define $S(j)$ as the number of packets correctly received during the time slot j. Hence, when the satellite processes a time slot, the load control algorithm runs the operations highlighted in the figure 3.

Input: $\widehat{G}(k_{Last})$, $p(k_{Last})$, $k_{\min}(k_{Last})$, $k_{\text{spread}}(k_{Last})$ and $S(k_{Last})$

Update of $\widehat{G_B}$ in future time slots

For $l = 0$ **to** $l = k_{spread}(k_{Last}) - 1$ **do**

$\widehat{G_B}\left(j + k_{\min}(j) + l\right)$ += $\frac{\widehat{G}(j)\left(\frac{1}{p(j)} - 1\right)}{k_{\text{spread}}(j)}$

End

Computation of $\widehat{G_R}$ in future time slots

Note: The '2' comes from $2t_{Slot}$*, see figure 1.*

$\widehat{G_R}(k_{Last} + k_{\text{Phy}} + k_{\text{MAC}} + 2) = \widehat{G}(k_{Last}) - S(k_{Last})$

Computation of $\widehat{G_N}$ in time slot k_{Last}

$\widehat{G_N}(k_{Last}) = \frac{\widehat{G}(k_{Last})}{p(k_{Last})} - \widehat{G_B}(k_{Last}) - \widehat{G_R}(k_{Last})$

Fig. 3: Operations runs by the load control mechanism on the informations from time slot k_{Last}. The load control mechanism realizes this operation not only on time slot k_{Last}.

Thanks to the algorithm detailed in figure 3, the load control mechanism knows $\widehat{G_R}\left(k_{\text{Estim}}\right)$ and $\widehat{G_B}\left(k_{\text{Estim}}\right)$. Then, the load control mechanism computes $\widehat{G_N}\left(k_{\text{Estim}}\right)$, two solutions are considered:

- The average method where $\widehat{G_N}$ values are averaged over a finite window of size S_w which leads to (3).

$$\widehat{G_N}\left(k_{\text{Estim}}\right) = \frac{\sum_{i=0}^{S_w - 1} \widehat{G_N}\left(k_{\text{Last}} - i\right)}{S_w} \tag{3}$$

- The exponential smoothing method [15]. In this method, the estimations are given according to the following function $F(\{x\})$.

$$\begin{cases} F(\{x\})_0 = x_0 \\ F(\{x\})_n = \alpha x_n + (1-\alpha) F(\{x\})_{n-1}, n > 0 \end{cases}$$

Where α belongs to $[0; 1]$ and $\{x\}$ represents the data sequence to predict in the future. This leads to equation (4).

$$\widehat{G_N}\left(k_{\text{Estim}}\right) = F(\{\widehat{G_N}\})_{k_{\text{Last}}} \tag{4}$$

Finally, thanks to (3) or (4), the load control mechanism computes $\widehat{G_T}\left(k_{\text{Estim}}\right)$. Two rules must be followed to compute accurately the expected load:

$$t_{min} > t_{\text{Phy}} + t_{\text{LC}} + t_{\text{slot}} \text{ and } t_{\text{MAC}} \geq t_{\text{LC}} \tag{5}$$

Indeed, devices blocked by the load control or devices that failed a transmission must not try to retransmit too quickly to be correctly counted by the load control mechanism. These rules are due to the satellite onboard processing duration which may be understood thanks to figures 1 and 2.

D. Computation of the access probability for the SS-MF-CRDSA

Suppose that the satellite knows exactly the load G_T. The satellite must apply an access probability that maximizes the throughput on the random access channel; this probability is computed in function of G_T.

1) Setting up equations:

- p is the access probability broadcasted by the satellite.
- $Pr\left(G = j | G_T = M\right)$ is the probability of having j packets transmitted while there were M devices which realized the access test. Since devices are independent, Pr is written as follows:

$$Pr\left(G = j | G_T = M\right) = \binom{M}{j} p^j (1-p)^{M-j}$$

- $S(j)$ represents the number of packets correctly transmitted on the RACH (i.e. throughput) when j packets are transmitted during the time slot; it is given by [13].
- $\overline{S}(p, G_T)$ is the average RACH throughput when the access probability p is applied to the expected load G_T.

We deduce the expression of $\overline{S}$:

$$\overline{S}\left(p, G_T = M\right) = \sum_{j=0}^{M} Pr\left(G = j | G_T = M\right) S\left(j\right) \tag{6}$$

To obtain the optimal access probability, we have to solve (7).

$$p_{Opt}\left(G_T\right) = \underset{p}{arg\,max}\ \overline{S}\left(p, G_T\right) \tag{7}$$

Since the mathematical expression of S is not known, the equation (7) cannot be resolved analytically. We introduce in the following a formula to approach the solution of (7).

2) Approximate solution: An approximation of the optimal access probability (p_{Opt}) is given by equation (8) where $G_m = \underset{G}{arg\,max}\ S(G)$.

$$p_{\text{Opt}}\left(G_T\right) = \begin{cases} \frac{G_m - 0.60\sqrt{\frac{G_m}{G_T}\left(G_T - G_m\right)}}{G_T} & \text{if } G_T > G_m \\ 1 & \text{else} \end{cases} \tag{8}$$

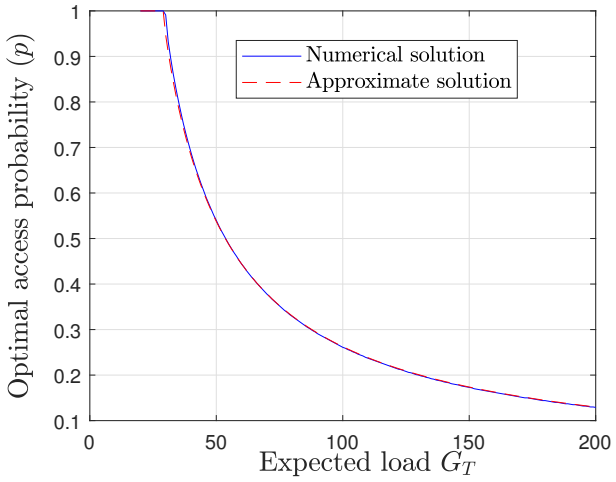

Fig. 4: Numerical solution vs Approximated solution (8).

In figure 4, we notice that the approximate solution is close to the numerical solution. The approximate solution can be used by the load control mechanism to compute the optimal access probability.

3) The rationale of the approximate solution: This part explains the idea behind (8). The physical layer has a waterfall effect (an abrupt drop in the throughput S when $G > G_m$) due to the utilization of successive interference cancellation by the receiver.

Hence, the access probability $p = \frac{G_m}{G_T}$ is not optimal since in half the cases (because of the binomial distribution) the number of packets transmitted to the satellite is greater than G_m which leads to a low number of packets correctly received because of the waterfall effect. The access probability shall be shifted to the left (i.e. $p < \frac{G_m}{G_T}$) to avoid as much as possible the waterfall area, see figure 5.

The shift is linked to the spatial dispersion of the binomial which leads to the following formula:

$$p = \frac{G_m - \gamma\sqrt{\frac{G_m}{G_T}\left(G_T - G_m\right)}}{G_T}$$

Then, the parameter γ is found by simulation, it is the parameter which approaches the most the numerical solution. Note that γ value depends on the slope of the throughput curve S before and after $G = G_m$.

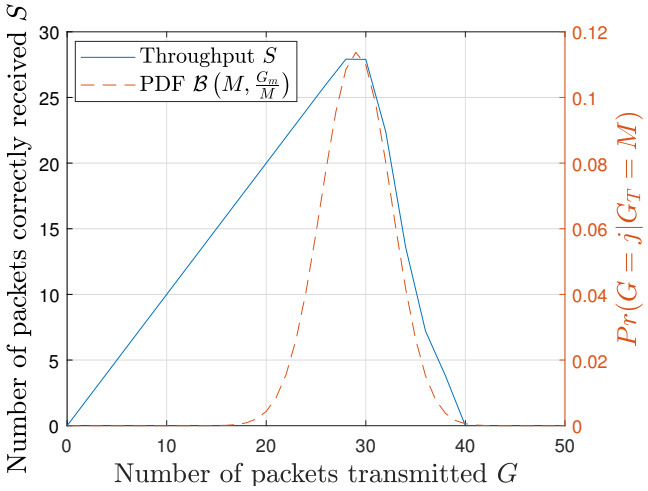

Fig. 5: Physical layer throughput (S) and the Probability Density Function (PDFs) of the distribution $\mathscr{B}\left(M, \frac{G_m}{M}\right)$ where $M = 50$.

E. Barring time parameters computation

The purpose of barring time parameters is to smooth overload over time. The calculation of these parameters is very critical. Indeed, the parameters should not be too small to prevent blocked devices to be delayed during an overloaded time slot which is detrimental for them since devices have a limited load control blocks ($N_{\text{Max Block}}$).

We consider that the satellite computes N_{OC}, the number of consecutive time slots since the load control is activated (i.e. when $p < 1$). N_{OC} is reset when the load control is disabled (i.e. $p = 1$) during $N_{\text{Wnd reset}}$ consecutive time slots. $N_{\text{Wnd reset}}$ is used to ensure the true end of the overload before resetting N_{OC}.

The first barring time parameter t_{min} is computed as:

$$t_{\text{min}} = \max\left(t_{Phy} + t_{LC} + 2t_{slot}, \left\lceil N_{OC}\frac{\widehat{G_T}}{G_m}\right\rceil t_{slot}\right) \quad (9)$$

In (9), t_{min} is function of N_{OC} and G_T, the objective is to adapt this parameter to different traffic scenarios. The link between t_{min} and N_{OC} allows to adapt to different overload durations. Thanks to its relationship with G_T, t_{min} could be adapted to various levels of overload and accordingly N_{OC} parameter allows t_{min} optimization for different overload duration.

And, the relationship with G_T allows adjusting to various levels of overload. The second barring time parameter t_{spread} is computed as follow:

$$t_{\text{spread}} = \left\lceil \widehat{G_T}\left(1 - p_{Opt}\left(\widehat{G_T}\right)\right) t_{slot}\right\rceil \quad (10)$$

In (10), t_{spread} is associated to the expected number of devices blocked by the load control $\widehat{G_T}\left(1 - p_{Opt}\left(\widehat{G_T}\right)\right)$.

V. Performance Evaluation

A. Simulation context

1) Traffic scenarios: We have created our traffic scenarios since no scenarios are available within the literature to evaluate the performance of load control mechanisms used by a satellite IoT system.

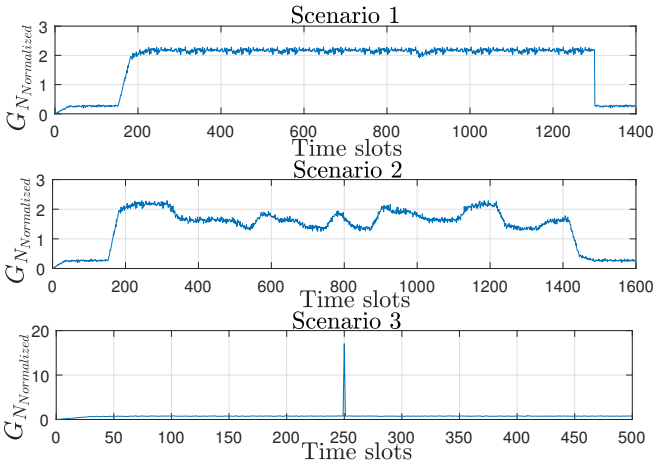

Fig. 6: Traffic profiles, $G_{N_{\text{Normalized}}}$ variation over time slots.

Figure 6 represents the traffic scenarios, $G_{N_{\text{Normalized}}}$ is the ratio between the number of new devices per time slot G_N and the maximum throughput of the random access channel S_{max}. Hence, when $G_{N_{\text{Normalized}}} > 1$, the number of new devices is superior to the maximal throughput achievable on the RACH, this leads to the RACH overload. The first scenario is basic; the overload occurs during a finite time window. This scenario could be associated with the busy hours when many sensors transmit their periodic report for instance. The second scenario is similar to the first one, except that, during the overload, $G_{N_{\text{Normalized}}}$ varies significantly. This scenario is useful to ensure that the load control mechanism adapts correctly to the load variations. Finally, the third scenario is a

very short overload peak, this peak can be caused by sensors exception reports (e.g. a power outage) or the simultaneous transmission of many sensors(e.g. from a wind farm). This scenario aims to ensure that the estimation of the expected load G_T is not deregulated by the peak.

2) Metrics: To evaluate the performance we consider metrics focused on the satellite and devices. The satellite metric is the number of packets correctly received by satellite per time slot (S), also known as the throughput. This metric is observed when the load control is activated to study the throughput loss due to the overload.

Metrics focused on devices are computed on devices whose first transmission attempt occurs when $G_{N_{\text{Normalized}}} > 1$. The metrics are the following:

- The probability of successful transmission $P_{Success}$.
- The number of transmissions NT.
- The number of transmissions blocked by the load control NB.

Metrics NT and NB are useful to have an idea of the energy consumed by devices to correctly transmit a packet.

3) Oracle Mode: We define a load control mode named the Oracle. With this mode, the satellite knows exactly the expected load by time slot i.e. $\widehat{G_T} = G_T$. Hence, the satellite applies the optimal access probability defined by (8) and appropriate barring time parameters. This mode is used to evaluate load control mechanisms performance since it leads to the performance upper boundary.

4) Simulation parameters: Table II summarizes parameters values used for performance evaluation.

N_{rep}	2
N_F	20
S_F	2
$N_{\text{Max Transmission}}$	10
$N_{\text{Max Block}}$	5
$N_{\text{Wnd reset}}$	5
t_{Phy}	15 t_{Slot}
t_{MAC}	5 t_{Slot}
t_{LC}	3 t_{Slot}

TABLE II: Default settings used during simulations.

B. Results simulations

The simulation results are divided into two parts. The first part is an overview of the load control mechanism performance while the second is focused on the impact of S_w, α, and β on the performance. In the following the load control mechanism named "A" uses the average method while the one named "ES" uses the exponential smoothing method.

1) Performance overview: We compare our load control mechanisms with the Oracle mode, and the Oracle mode without barring time parameters (i.e. $t_{\text{min}} = t_{Phy} + t_{LC} + 2t_{slot}$, $t_{\text{spread}} = 0$ and, $N_{\text{Max Block}} = \infty$). This last mode refers to the load control mechanism [11] and [12], which do not consider barring time parameters.

Table III describes the performance when the load control mechanism uses a perfect RACH load estimator ($\beta = 0$). We notice that our load control mechanism reach astonishing performance close to the Oracle mode. The table demonstrates the benefit of using barring time parameters. Without them, the number of load control blocks is high which leads to an energy over-consumption. Hence the interest of our mechanism to tackle the overload issue on random access.

Scenarios	LC type	$P_{Success}$	NB	NT	$\frac{\overline{S}}{S_{max}}$
1	A($S_w = 40$)	99.94 %	0.93	1.07	87.36 %
	ES($\alpha = 0.05$)	99.95 %	0.92	1.07	87.60 %
	Oracle	99.99 %	0.79	1.01	92.32 %
	Oracle w/o barring time	100 %	55.58	1.04	89.45 %
2	A($S_w = 40$)	99.84 %	0.82	1.08	83.70 %
	ES($\alpha = 0.05$)	99.84 %	0.80	1.07	84.35 %
	Oracle	99.99 %	0.72	1.01	92.43 %
	Oracle w/o barring time	100 %	42.38	1.04	89.51 %
3	A($S_w = 40$)	100 %	0.96	2.92	83.79 %
	ES($\alpha = 0.05$)	100 %	0.96	2.00	83.64 %
	Oracle	100 %	0.95	1.00	98.44 %
	Oracle w/o barring time	100 %	19.54	1.04	91.6 %

TABLE III: Performance overview

2) Influence of S_w, α, and β on the performance: We focus the study on the traffic scenario 1 and the metric $P_{Success}$ since the conclusions are similar to the others scenarios and metrics.

Thanks to both figures 7 and 8, we notice the impact of β on the performances. The transmission reliability ($P_{Success}$) collapses when β rises since the RACH load estimation are less accurate which impacts the estimation of the expected load. But we remark that even with an inaccurate RACH load estimator ($\beta > 0.3$), we can find a configuration of our load control mechanism enabling to obtain satisfactory performances.

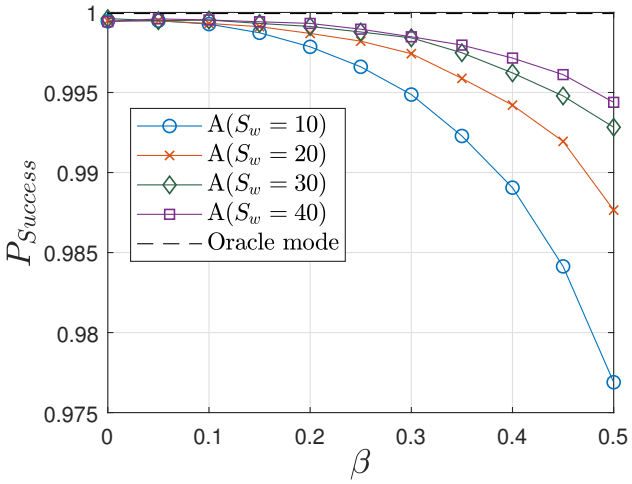

Fig. 7: Influence of β and S_w on the mechanism A performance with traffic profile 1.

Figure 7 shows also the impact of S_w. Better performance is found out when S_w rises since with a larger window

the average is more accurate (see (3)). The performance gain is reduced as we increase S_w since the average is larger enough. Moreover, the utilization of a large window ($S_w > 40$) is not able to catch the traffic variation accurately. There is a trade-off to find between the average convergence and the load control mechanism reactivity.

Figure 8 describes the impact of α on the performance of load control mechanism based on the exponential smoothing method. Better performance is found out when α is low since the RACH load estimations are more smoothed. Hence, the traffic trend is better known by the load control mechanism.

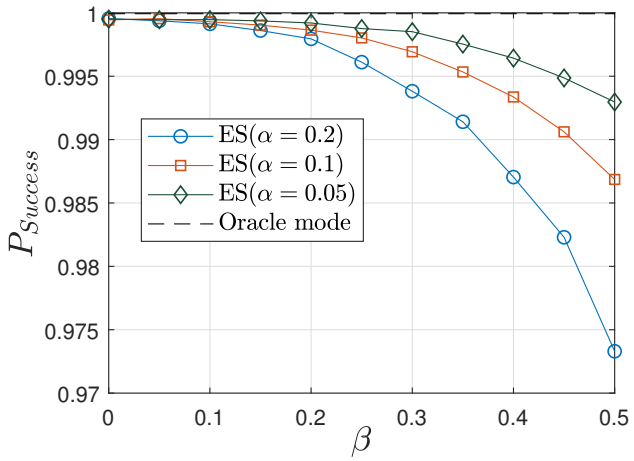

Fig. 8: Influence of β and α on the mechanism ES performance with traffic profile 1.

VI. Conclusion

In this paper, we have tackled the overload on time slotted random access channel using the SS-MF-CRDSA, a random access method made for IoT satellite systems. The contributions of the papers are the following:

- We have defined two versions of an algorithm which estimates the load on the random access channel during future time slots.
- We have calculated the optimal access probability (8) for a random access channel using the SS-MF-CRDSA. We note that the method used to compute (8) may be used to compute the optimal access probability of other random access methods.
- We define a method to calculate the barring time parameters dynamically, see equations (9) (10). We have shown (see table III) that the consideration of the barring time parameters is essential to avoid devices' energy over-consumption.

Furthermore, our work can be extended to all random access channel with a downlink to carry signaling information such as the load control parameters.

As prospective work, we could extend the performance evaluation to the transmission delay not studied in this paper. Regarding the IoT, we notice a wide variety of transmission delay requirements. Critical applications like power grid sensors or Tsunami alarm need short transmission delay while basic sensors may accommodate a transmission delay of several minutes or hours. It would be interesting to explore the overload with both types of applications and to prioritize the transmission of delay-sensitive applications.

We could also extend the study with another access method for the random access channel. The first impact will be the modification of the optimal access probability formula. It will be interesting to observe if the method to compute the barring time parameters is still efficient with another access method.

We have observed the positive impact of the barring time parameters on the performance. It would be interesting to optimize the computation of these latter parameters regarding the transmission reliability, the transmission delay and the number of load control blocks.

References

[1] Gartner Cabinet, "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016", available at http://www.gartner.com/newsroom/id/3598917, Feb. 2017.

[2] Gartner Cabinet, "Gartner Cabinet forecasts for IoT by 2020", available at www.gartner.com/newsroom/id/3436717, Sept. 2016.

[3] 3GPP TR 45.820 V13.1.0, "Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT)", Nov. 2015.

[4] E. Soltanmohammadi, K. Ghavami and M. Naraghi-Pour, "A Survey of Traffic Issues in Machine-to-Machine Communications Over LTE," in IEEE Internet of Things Journal, vol. 3, no. 6, pp. 865-884, Dec. 2016.

[5] Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System; Part 2: Lower Layers for Satellite standard ETSI EN 301 545-2 V1.1.1 (2012-01).

[6] 3GPP, TR 23.898, "Access Class Barring and Overload Protection", V7.0.0, Mar. 2005

[7] U. Phuyal, A. T. Koc, M. H. Fong and R. Vannithamby, "Controlling access overload and signaling congestion in M2M networks," 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, 2012, pp. 591-595.

[8] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset and V. Casares-Giner, "Performance analysis of access class barring for handling massive M2M traffic in LTE-A networks," 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-6.

[9] 3GPP TR 37.868 V11.0.0, "Study on RAN Improvements for Machine Type Communications", Sep. 2011.

[10] C. Kalalas and J. Alonso-Zarate, "Reliability analysis of the random access channel of LTE with access class barring for smart grid monitoring traffic," 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 2017, pp. 724-730.

[11] H. Jin; W. Toor; B. C. Jung; J. B. Seo, "Recursive Pseudo-Bayesian Access Class Barring for M2M Communications in LTE Systems," in IEEE Transactions on Vehicular Technology , vol.PP, no.99, pp.1-1.

[12] S. Duan, V. Shah-Mansouri, Z. Wang and V. W. S. Wong, "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks," in IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9847-9861, Dec. 2016.

[13] A. Mengali; R. De Gaudenzi; P. D. Arapoglou, "Enhancing the Physical Layer of Contention Resolution Diversity Slotted ALOHA," in IEEE Transactions on Communications , vol.PP, no.99, pp.1-1

[14] E. Casini, R. De Gaudenzi and O. Del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks," in IEEE Transactions on Wireless Communications, vol. 6, no. 4, pp. 1408-1419, April 2007.

[15] Robert G. Brown, Richard F. Meyer and D. A. D'Esopo, "The Fundamental Theorem of Exponential Smoothing",Operations Research Vol. 9, No. 5 (Sep. - Oct., 1961), pp. 673-687