Random Access in DVB-RCS2: Design and Dynamic Control for Congestion Avoidance

Alessio Meloni, Student Member, IEEE, Maurizio Murroni, Senior Member, IEEE

Abstract—In the current DVB generation, satellite terminals are expected to be interactive and capable of transmission in the return channel with satisfying quality. Considering the bursty nature of their traffic and the long propagation delay, the use of a Random Access technique is a viable solution for such a MAC scenario. In this paper Random Access communication design in DVB-RCS2 is considered with particular regard to the recently introduced Contention Resolution Diversity Slotted Aloha (CRDSA) technique. The paper presents a model for design and tackles some issues on performance evaluation of the system by giving intuitive and effective tools. Moreover, dynamic control procedures able to avoid congestion at the gateway are introduced. Results show the advantages brought by CRDSA on DVB-RCS2 with regard to the previous state of the art.

Index Terms—Retransmission, Successive Interference Cancellation, Slotted Aloha, Random Access, Congestion Avoidance, Dynamic Control, Satellite Broadcasting, Interactive Terminals

I. INTRODUCTION

Current DVB standards are increasingly going towards the need of high-speed bidirectional networks for consumer interactivity [1]. Nevertheless in consumer type of interactive satellite terminals, users generate a large amount of low duty cycle and bursty traffic with frequent periods of inactivity in the return link. Under these operating conditions, the traditionally used Demand Assignment Multiple Access (DAMA) satellite protocol does not perform optimally, since the response time for the transmission of short bursts can be too long. For this reason, in the recently approved specification for the next generation of Interactive Satellite Systems (DVB-RCS2) [2], the possibility of sending logon, control and even user traffic using Random Access (RA) in timeslots specified by the Network Control Center (NCC) is provided to Return Channel Satellite Terminals (RCST). In particular, two methods are considered for RA: the first one is Slotted Aloha (SA) [3] [4], the second is called Contention Resolution Diversity Slotted Aloha (CRDSA) [5]. SA represents a well established RA technique for satellite networks in which users send their bursts within slots in a distributed manner, i.e. without any central entity coordinating transmission. This allows to reach an average throughput around $36 \text{ [packets/slot]}$ despite the possibility of collision among bursts from different users. Nevertheless, in practice SA works with very moderate channel load to ensure acceptable delay and loss probability $(PLR = 10^{-3})$. This gap has been recently filled with the introduction of a new technique named CRDSA that is able to boost the throughput even up to values close to $1\text{[packets/slot]}$. This technique is based on the transmission of a chosen number of replicas for each burst payload as in Diversity Slotted Aloha (DSA) [6]. However, differently from DSA, in CRDSA each burst copy has a pointer to the location of the other replicas of the same burst payload so that when the frame arrives at the receiver a Successive Interference Cancellation (SIC) process is accomplished. The SIC process consists in removing the content of already decoded bursts from the slots in which collision occurred. Doing so, it is possible to restore the content of bursts that had all their replicas colliding thus boosting the throughput. Nevertheless the use of interference cancellation can benefit from power unbalance among different sources [7] [8].

The specification for the lower layers of DVB-RCS2 [2] states that the applications using the interactive network may rely on network internal contention control mechanisms to avoid sustained excessive packet loss resulting from simultaneous destructive transmissions (i.e. congestion). For this reason, the definition of such a mechanism is claimed to be out of scope for the document. In this paper, a complete model for the design of such a system and simple yet effective dynamic retransmission policies aiming at avoiding congestion at the gateway are presented. To do so, we rely on the analysis we carried out in [9], in which the model was firstly introduced and throughput results for CRDSA with geometrically distributed retransmissions were analyzed. The presented model is based on the definition of Equilibrium Contour [10] and Channel Load Line. The Equilibrium Contour represents a set of points for different load values, for which the ongoing communication is in equilibrium in the sense that the number of newly transmitted packets is equal to the number of packets successfully sent so that the total number of packets “queuing for retransmission” do not change. The Channel Load Line completes this analysis telling which of these equilibrium points are actually of equilibrium for a given scenario. After revising the model presented in [9], tools for the computation of some important parameters such as throughput, packet delay and the First Exit Time are given. Furthermore, design guidelines able to ensure that the communication is taking place in a desirable and stable manner are discussed. Finally, dynamic procedures of the control limit type [11] are introduced as a mean to avoid congestion in DVB-RCS2 when using RA and in particular CRDSA as transmission mode. The paper concludes with a thorough comparison between the use of dynamic control policies in SA and CRDSA and with the extension of the presented model to the case of infinite...
population, that represents a typical DVB-RCS2 scenario in which a big number of terminals are part of the interactive network and can manage multiple packet transmissions.

II. CRDSA SYSTEM OVERVIEW

Consider a multi-access channel populated by a total number of users $M$. Users are synchronized so that the channel is divided into slots and $N_s$ consecutive slots are grouped in a frame. The probability that at the beginning of a frame an idle user will send a new packet is $p_0$. When a frame starts, users willing to transmit place $d$ copies of the same packet over the $N_s$ slots of that frame. The number of copies $d$ can be either the same for each packet or not. The first case is known as Constant Replication CRDSA (CR-CRDSA); the second is known as Variable Replication CRDSA (VR-CRDSA). In this second case the number of copies is determined according to a predefined packet copies’ probability distribution. Packet copies are nothing else than redundant replicas except for the fact that each one contains a pointer to the location of the others. These pointers are used in order to attempt restoring collided packets at the receiver by means of SIC, i.e. by removing the interfering content of already decoded packets thanks to their location’s knowledge. Nevertheless, the physical structure of the burst copies used is equal to the one used for dedicated access with respect to the burst construction, the waveform, the timeslot structure and the burst reception.

Throughout the paper perfect SIC and ideal channel are assumed, which means that the receiver is always able to perfectly apply SIC without errors and no external sources of disturbance such as noise that might require SNR estimation [12] are present. Therefore, the only cause of unsuccessful decoding of packets is interference among them. Given these conditions, consider Figure 1 representing a practical example of SIC. Each slot can be in one of three states:

- no packet’s copies have been placed in a given slot, thus the slot is empty;
- only 1 packet’s copy has been placed in a given slot, thus the packet is correctly decoded;
- more than 1 packet’s copy has been placed in a given slot, thus resulting in collision.

In DSA, if all copies for a given packet collided the packet is surely lost. In CRDSA, if at least one copy of a certain packet has been correctly received (see User 4), the contribution of the other copies of the same packet can be removed from the other slots thanks to the knowledge of their location provided by the pointers contained in the decoded packet. This process might allow to restore the content of packets that had all their copies colliding (see User 2) and iteratively other packets may be correctly decoded up to a point in which no more packets can be restored (see User 1 and 3) or until the maximum number of iterations $I_{max}$ for the SIC process is reached, if a limit has been fixed.

If a packet has not been successfully decoded at the end of the SIC process and a retransmission request is sent by the NCC, a feedback is created between the transmitter and the receiver side. This could cause the channel to get to a state of congestion in which the load is too big and causes excessive packet loss due to simultaneous destructive transmissions of the terminals in the return channel. For this reason, a careful design of the communication based on a stability model and the possible application of control policies able to avoid congestion are of great interest for practical DVB-RCS applications.

III. STABILITY MODEL

Consider the RA communication system presented in Section II. Each user can be in one of two states: Thinking (T) or Backlogged (B). Users in T state generate a new packet for transmission with probability $p_0$ over a frame interval; if so, no other packets are generated until successful transmission for that packet has been acknowledged. Users in B state have unsuccessfully transmitted their packet and keep attempting to retransmit it with probability $p_r$ over a frame interval. In the followings, we assume that users are acknowledged about the success of their transmission at the end of the frame in which the packet has been transmitted (i.e. immediate feedback). Nevertheless this constraint will be relaxed in the last sections.

Defining

- $N_B^f$: backlogged users at the end of frame $f$
- $G_B^f = \frac{N_B^{(f-1)}p_r}{N_s}$: expected channel load of frame $f$ due to users in B state
- $G_T^f$: expected channel load of frame $f$ due to users in T state
- $G_{IN}^f = G_T^f + G_B^f$: expected total channel load of frame $f$
- $PLR^f(G_{IN}^f, N_s, d, I_{max})$: expected packet loss ratio of frame $f$
- $G_{OUT}^f = G_{IN}^f(1 - PLR(G_{IN}^f, N_s, d, I_{max}))$: part of load successfully transmitted in frame $f$, i.e. throughput.

the equilibrium contour can be written as

$$G_T = G_{OUT} = G_{IN}(1 - PLR(G_{IN}, N_s, d, I_{max}))$$

that represents the locus of points for which at any time the expected channel load due to users in T state is equal to the expected throughput $G_{OUT}$. Notice that the frame number $f$ has been omitted, since in equilibrium state this condition is
expected to hold for any frame. From the definition above we can also gather that the expected number of backlogged users remains the same frame after frame. Therefore

\[ N_B = \frac{G_{IN} \text{PLR}(G_{IN}, N_s, d, I_{max}) N_s}{p_r} \]  

Equations (1) and (2) completely describe the equilibrium contour on the \((G_T, N_B)\) plane for different values of \(G_{IN}\). Notice that the equilibrium contour itself is an infinite set of equilibrium points. In fact, we have not considered \(M\) and \(p_0\) so far. This two essential parameters constitute the channel load line and provide the knowledge of the actual stability points for a certain scenario.

Consider \(M\) and \(p_0\) to be constant (i.e. stationary input). Given a certain number of backlogged packets, the channel load line expresses the expected value of channel load due to users in \(T\) state. For the finite population case, the channel load line can be defined as

\[ G_T = \frac{M - N_B}{N_s} - p_0 \]

while for \(M \to \infty\) the channel input can be described as a Poisson process with expected value \(\lambda\) [thinking users] \([4]\) so that

\[ G_T = \frac{\lambda}{N_s} \]

for any \(N_B\) (i.e. the expected channel input is constant and independent of the number of backlogged packets).

Figure 2 shows the equilibrium contour and the channel load line for various cases. The equilibrium contour divides the \((N_B,G_T)\) plane in two parts and each channel load line can have one or more intersections with the equilibrium contour. These intersections are referred to as equilibrium points since \(G_{OUT} = G_T\). The rest of the points of the channel load line belong to one of two sets: those on the left of the equilibrium contour represent points for which \(G_{OUT} > G_T\), thus situations that yield to decrease of the backlogged population; those on the right represent points for which \(G_{OUT} < G_T\), thus situations that yield to growth of the backlogged population.

Therefore we can gather that an intersection point where the channel load line enters the left part for increasing backlogged population corresponds to a stable equilibrium point. In particular, if the intersection is the only one, the point is a globally stable equilibrium point (indicated as \(G_T^G, N_B^G\)) while if more than one intersection is present, it is a locally stable equilibrium point (indicated as \(G_T^S, N_B^S\)).

If an intersection point enters the right part for increasing backlogged population, it is said to be an unstable equilibrium point (indicated as \(G_T^U, N_B^U\)).

Figure 2(a) shows a stable channel. The globally stable equilibrium point can be referred as channel operating point in the sense that we expect the channel to operate around that point. With the word around we mean that due to statistical fluctuations, the actual \(G_T\) and \(G_B\) (and thus also \(G_{IN}\) and \(G_{OUT}\)) may differ from the expected values. In fact, the actual values have binomially distributed probability for \(G_B\) and \(G_T\) with finite \(M\) and Poisson distributed probability for \(G_T\) with infinite \(M\). Nevertheless, the assumption of taking the expected value has been already validated in [9].

Figures 2(b) and 2(c) show two unstable channels respec-
tively for finite and infinite number of users. Analyzing this two figures for increasing number of backlogged packets, the first equilibrium point is a stable equilibrium point. Therefore the communication will tend to keep around it as for the stable equilibrium point in Figure 2(a) and we can refer to it once again as channel operating point. However, due to the abovementioned statistical fluctuations, the number of backlogged users could pass the second intersection and start to monotonically increase. In the case of finite logged users could pass the second intersection and start to abovementioned statistical fluctuations, the number of backlogged users could pass the second intersection and start to monotonically increase. In the case of finite M, this increment goes on till a new intersection point is reached, while in the case of infinite M the expected number of backlogged users increases without any bound. In the former case, this third intersection point is another stable equilibrium point known as channel saturation point, so called because it is a condition in which almost any user is in B state and G_{OUT} approaches zero. In the latter case, we can say that a channel saturation point is present for N_B → ∞. Finally Figure 2(d) shows the case of an overloaded channel. In this case there is only one equilibrium point corresponding to the channel saturation point. Therefore, even though the channel is nominally stable, the point of stability occurs in a non-desired region. For this reason this case is separated and distinguished from what is intended in this work as stable channel.

IV. PACKET DELAY MODEL

Assuming a channel at its channel operating point, we would like to know what is the expected distribution and average delay associated to packets that are successfully received at the gateway. This can be described using a discrete-time Markov chain with the two states B and T (Figure 3). The edges emanating from the states represent the state transitions occurring to users which depend on p_r and the expected PLR at the channel operating point. If a packet transmitting for the first time receives a positive acknowledgment the user stays in T state while in case of negative acknowledgment the user switches to B state until successful retransmission has been accomplished. We assume a frame duration to be the discrete time unit of this Markov chain. Therefore the packet delay D_{pkt} is calculated as the number of frames that elapse from the beginning of the frame in which the packet was transmitted for the first time, till the end of the one in which the packet was correctly received.

\[ Pr\{D_{pkt} = f\} = \begin{cases} 1 - PLR, & \text{for } f = 1 \\ PLR [p_r (1 - PLR)] \cdot \left[1 - p_r + PLR p_r \right]^{f-2}, & \text{for } f > 1 \end{cases} \]

which results in the average delay

\[ Av[D_{pkt}] = \sum_{f=1}^{\infty} f \cdot Pr\{D_{pkt} = f\} \]

Equation 6 can alternatively be rewritten in a simpler and more intuitive form by means of Little’s Theorem [10] [14] considering that the average number of backlogged users in a stable system is equal to the average time spent in backlogged state, multiplied by the arrival rate of new packets G_T (that we know to be equal to G_{OUT} at the operational point). Therefore

\[ Av[D_{pkt}] = \frac{N_B}{G_{OUT} \cdot N_s} \]

where the presence of N_s in the formula has the aim of normalizing the delay to the frame unit.

<table>
<thead>
<tr>
<th>p_r</th>
<th>Av[D_{pkt}]_ana</th>
<th>Av[D_{pkt}]_sim</th>
<th>PLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>543.48</td>
<td>541.89</td>
<td>0.99816</td>
</tr>
<tr>
<td>0.4</td>
<td>3.17</td>
<td>3.16</td>
<td>0.4657</td>
</tr>
<tr>
<td>0.2</td>
<td>2.01</td>
<td>2.12</td>
<td>0.108</td>
</tr>
<tr>
<td>0.05</td>
<td>2.53</td>
<td>2.66</td>
<td>0.0013</td>
</tr>
<tr>
<td>0.005</td>
<td>5.08</td>
<td>5.22</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table I

Analytic and simulation average packet delay at the channel operating point for CRDSA with N_s = 100 slots, I_{max} = 20, M = 350 and p_0 = 0.18.

Figures 4 and 5 show some results based on the example of overloaded channel given in Section III. In particular p_r is progressively decreased in order to evaluate how distribution and average packet delay change. Figure 4 represents the channel load line and equilibrium curves for several p_r values. When p_r is decreased, the equilibrium contour moves upwards and the point of equilibrium is found for bigger values of throughput while N_B decreases. Therefore, from Little’s theorem, it is intuitive to expect that users will spend less time in B state. However, if we keep decreasing p_r after the throughput peak is reached, the throughput starts to decrease while the number of backlogged users increases again thus yielding to increase of the average packet delay. This is confirmed by the results given in Table I. Figure 5 shows that while having a small p_r is beneficial for the PLR, it also means that users in B state will generally wait longer before retransmitting a packet. As a consequence, the smaller the p_r value the less steep the cumulative distribution curve is, so that at a certain point curves with bigger p_r overcome it in terms of probability to receive a packet before a certain time deadline. Finally the simulated average packet delay associated to the examples in Figure 5 is reported in Table I together with the expected results, demonstrating the validity of the model.
thinking users have transmitted and $M$ of thinking users each j.

The probability to pass from a number of backlogged users

\[ \begin{align*}
&= 0.005 \\
&= 0.05 \\
&= 0.2 \\
&= 0.4 \\
&= 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 \end{align*} \]

Figure 5. Packet delay cumulative distribution for CRDSA with $N_s = 100$ slots, $I_{max} = 20$, $M = 350$ and $p_0 = 0.18$.

V. FIRST EXIT TIME

In Section III the concept of unstable channel has been introduced. The First Exit Time (FET) allows us to quantify how much the channel is unstable in terms of time needed to exit the stability region. To calculate this parameter we consider a discrete-time Markov chain which uses the number of backlogged users $N_B$ as describing state, since equilibrium points and more generally the state of the communication can be uniquely identified with it. Let us define $P$ as the transition matrix for this Markov chain, with each element $p_{ij}$ denoting the probability to pass from a number of backlogged users $N_B = j$ to a number of backlogged users $N_B = i$. To calculate each $p_{ij}$ value we need to know the probability that having a certain number of backlogged users $j$ and a certain number of thinking users $M - j$, respectively $b$ backlogged users and $t$ thinking users have transmitted and $s = t - (i - j)$ out of

\[ p_{ij} = \sum_{t=0}^{M-j} \sum_{b=0}^{M} \binom{M-j}{t} p_0^t (1 - p_0)^{M-j-t} \cdot \binom{j}{b} p_r^b (1 - p_r)^{j-b} \cdot q(s | t + b) \]  \hspace{1cm} (8) \]

where $q(s | t + b)$ is the probability that $s$ packets are correctly decoded if $t + b$ packets are transmitted\(^2\). Let us then call $Q$ the matrix with elements $q_{s,t+b}$ corresponding to probability $q(s | t + b)$.

In Figure 6 the resulting P matrix is shown. Accordingly to the bidimensional representation used in Section III, we can see here that there are two peaks corresponding to the two expected locally stable equilibrium points. However, it is more fruitful for our considerations to look at this graph from a plane perspective plotting a line for a $(i,j)$ plane perspective plotting a line for $i = j$. Points below this line represent decrements of the number of backlogged packets while points above represent an increment. Non-zero probability points look like a knee close to the line $i = j$, highlighting that possible transitions are found only for small changes in the number of backlogged packets compared to the total population $M$. Taking single values of $j$, we can roughly identify 3 regions of the wake:

- for small values of $j$ (close to the channel operating point) non-zero probabilities are found both for increase and decrease of $N_B$;
- for values of $j$ close to $M$ (i.e. close to the channel saturation point) non-zero probabilities are found both for increase and decrease of $N_B$;

\(^2\)For the same reason claimed before for the PLR values used for the stability model, also $q(s | t + b)$ values are calculated from simulations over a big amount of runs (over $10^6$ per $(t + b)$ value) since analytical formulas able to tightly describe the probability for a packet to be correctly decoded have not been found yet. Nevertheless in the followings a method to reduce this computational effort is introduced.
for the rest of the $j$ values non-zero probabilities are almost exclusively found for $i > j$ so that the number of backlogged users monotonically increases.

Based on the $P$ matrix, it is now possible to calculate the probability $B_{i+1}^{f}$ to be in a certain state $N_B = i$ at frame $f+1$, given a certain probability distribution $B_f = [B_0^f \ B_1^f \ \cdots \ B_M^f]$ at frame $f$:

$$
\begin{bmatrix}
B_0^{f+1} \\
B_1^{f+1} \\
\vdots \\
B_M^{f+1}
\end{bmatrix} =
\begin{bmatrix}
p_{0,0} & \cdots & p_{0,M} \\
p_{1,0} & \ddots & \vdots \\
\vdots & & \ddots \\
p_{M,0} & \cdots & p_{M,M}
\end{bmatrix}
\begin{bmatrix}
B_0^f \\
B_1^f \\
\vdots \\
B_M^f
\end{bmatrix}
$$

Generalizing Equation 9, the state distribution at each frame $(f+1)$ can always be calculated from the initial probability distribution $B_0^0 = [B_0^0 \ B_0^1 \ \cdots \ B_0^M]$ and the matrix $P^{f+1}$. Therefore, it is now possible to calculate the average and cumulative distribution of the FET from a given initial state. In this example we consider as initial state $B_0^0 = [1 \ 0 \ 0 \ \cdots \ 0 \ 0]$.

Figure 7 reports the evolution frame after frame of the state probability for the $P$ illustrated in Figure 6. As we can see, when $f$ increases the probability of being around the locally stable point in the region of high throughput decreases while the probability of being in saturation increases.

The entire description of any state of this Markov chain is of great interest but most of the times impractical, mainly because the dimension of the $P$ matrix is equal to $M^2$ and $P$ is derived from simulations. Therefore it could take a huge computational effort to calculate it. Nevertheless it is of interest to know the probability that the stability region around the operational point has been left in a certain moment, regardless of the actual state once we enter the region that yields to saturation. In fact, it has been already pointed out in the description of Figure 6 that the further the communication gets from the first point of local stability, the smaller is the probability that the communication will come back to the stability region. For this reason, as a first approximation, it is possible to consider an absorbing state $N_B^{abs} = (N_B^U) + 1$ that groups all the probabilities for states from $(N_B^U) + 1$ to $M$. Doing so, our $P$ matrix can be rewritten as

$$
P =
\begin{bmatrix}
p_{0,0} & \cdots & p_{0,N_B^U} & 0 \\
\vdots & \ddots & \vdots & \vdots \\
p_{N_B^U,0} & \cdots & p_{1,N_B^U} & 0 \\
1 - \left(\sum_{\alpha=0}^{N_B^U} p_{\alpha,0}\right) & \cdots & 1 - \left(\sum_{\alpha=0}^{N_B^U} p_{\alpha,N_B^U}\right) & 1
\end{bmatrix}
\tag{10}
$$

The $P$ matrix in Equation 10 takes much less effort than calculating the entire $P$ matrix since only $(N_B^U + 1)^2$ elements need to be computed thus speeding up calculations. Considering that $\sum_{\alpha=0}^{N_B^U} p_{\alpha,j}$ must be equal to 1, the probability to get to the absorbing state $N_B^{abs}$ from a certain state $j$ can be computed as $p_{i,j} = 1 - (\sum_{\alpha=0}^{N_B^U} p_{\alpha,j})$. Moreover, since $N_B^{abs}$ is an absorbing state, any probability to leave this state is zero while the probability of staying in the absorbing state is equal to 1.

We now aim at verifying the validity of this approximation. Figure 8 shows the distribution for the analytic results with absorbing state and the simulated results without the approximation of the absorbing state. In particular, regarding simulations if $N_B$ crosses the value $(N_B^U)$ but it comes back to the stability region, the FET is reset and the next exit is waited to set it again. For the analytic results the FET is calculated accordingly to the reduced matrix in Equation 10. As already pointed out in the description of Figure 6, when the number of backlogged packets $N_B$ is still close to the unstable equilibrium point, a certain probability that the communication will come back from the instability region is present. This is the reason why in Figure 8 the two curves present a

![Figure 7. Evolution of $B_f$ for the $P$ matrix in figure 6.](image)

![Figure 8. FET cumulative distribution for CRDSA with $N_s = 100$, $I_{max} = 20$, $M = 350$, $p_0 = 0.18$, $p_r = 1$.](image)
notable difference. However, this problem can be solved by considering $N_{B}^{abs} = \lceil N_B \rceil + 1 + \Delta$ as absorbing state, where $\Delta$ is a positive integer big enough to ensure that the probability of false exits from the stability region (i.e. the probability that $N_B$ crosses the stability region but comes back to values $< N_{B}^{abs}$) is sufficiently low. Doing so, simulation and analytical results with the approximation of absorbing state perfectly match. Figure 9 shows how the difference among the two average FETs decreases while increasing the chosen value for $N_{B}^{abs}$. In conclusion, we can state that analytic results with the approximation of absorbing state represent a lower bound for the actual FET. Therefore during this kind of analysis a tradeoff between tightness of the results and simplicity of computation is present.

Finally, we claim that the model presented is also adaptable to the case of infinite population. In fact the probability transition matrix $P$ can be adapted by considering

$$p_{ij} = \sum_{t=0}^{\infty} \sum_{b=0}^{\infty} \frac{\lambda^t e^{-\lambda}}{t!} \cdot \binom{j}{b} p_{r}^{b} (1 - p_{r})^{N_B - b} \cdot q(s|t + b)$$

(11)

Although in this case $P$ has infinite size, it has been just shown that is possible and practical to reduce the $P$ matrix by means of an absorbing state. With this expedient, also this case can be treated since $P$ is transformed to an equivalent matrix with finite size.

VI. DESIGN SETTINGS

In this section we will resume the role of some crucial parameters such as $M$, $p_0$, $p_r$ and explain how the communication state changes when changing them and what are the relations to be considered when designing DVB-RCS2 communications in RA mode. In fact, since optimization depends on constraints and degrees of freedom of the particular study case, a deep investigation of the relation among these crucial parameters is needed in order to understand tradeoffs that have to be faced. Moreover, the same discussion is introductory to understand how control policies can help to obtain better performance and a stable channel even in case of statistical fluctuations that could yield to instability.

Consider Figure 10. The first thing we can notice is that $M$ and $p_0$ only influence the channel load line while changing $p_r$ corresponds to a change of the equilibrium contour. In Figure 10(a) we can see that $p_0$ influences the slope of the channel load line that becomes steeper when increasing $p_0$. This is intuitive to understand since for fixed $M$, if $p_0^A < p_0^B$ then $G_T^A < G_T^B$. Figure 10(b) shows that $M$ shifts the channel
values of $p_r$ (Figure 10(c)). As already shown, if the value $p_r$ is sufficiently small, it is possible to stabilize a channel initially unstable. This represents the ground base for one of the control policies shown in the next section. Concerning possible constraints, we can identify the followings:

1) stability
2) maximize $p_0$ or guarantee a value $p_0 \geq p_{0}^{\text{min}}$
3) maximize $M$ or guarantee a value $M \geq M^{\text{min}}$
4) minimize the delay or guarantee that a certain value $A_{\text{V}}[D_{\text{pkt}}]^{\text{max}}$ is not exceeded
5) maximize the throughput or guarantee a value $G_{\text{OUT}} \geq G_{\text{OUT}}^{\text{min}}$

For constraint 1) it has been previously shown that the stability corresponds to having a single intersection between the channel load line and the equilibrium curve (i.e. a single point of equilibrium). Constraints from 2) to 5) can be graphically represented on the $(G_T, N_B)$ plane by the following formulas:

2) \[ \frac{G_T \cdot N_s}{M - N_B} \geq p_{0}^{\text{min}} \]  \hspace{1cm} (12)
3) \[ \frac{G_T \cdot N_s}{p_0} + N_B \geq M^{\text{min}} \]  \hspace{1cm} (13)
4) \[ \frac{N_B}{G_T \cdot N_s} \leq A_{\text{V}}[D_{\text{pkt}}]^{\text{max}} \]  \hspace{1cm} (14)
5) \[ G_T \geq G_{\text{OUT}}^{\text{min}} \]  \hspace{1cm} (15)

In other words, each formula simply represents an admissible area where the operational point has to be, as illustrated in Figure 11. Any study case will be characterized by one or more degrees of freedom and one or more design constraints. In case of more than one design constraint, the resulting constraint will be the intersection of the respective admissible areas. Therefore we can conclude that the equilibrium contour not only represents the expected stability and throughput of the communication, but also its constraints. This allows a design that does not need any use of complicated formulas in order to understand feasible settings. Notice that the choice of $N_s$ has not been considered as a design parameter. This choice have basically two reasons: first of all in DVB-RCS2, is the NCC that decides how many slots must be allocated for RA; secondly, the best choice for $N_s$ can always be considered between 100 and 200 since a lower value would degrade too much the throughput performance while a bigger value would degrade too much the delay performance.

VII. PACKET DEGREE AND RA TECHNIQUE

In the previous sections we have concentrated on the best selection of $p_0$, $p_r$ and $M$ once the number of replicas per packet (packet degree) is chosen. However also the number of packet copies is a design parameter\(^3\). Figure 13 shows that up to moderate channel loads, a greater packet degree

\(^3\)In this paper we compare various packet degrees under the assumption of equal received power per replica. For discussion on the power efficiency when considering transponder’s average power limits refer to [15].
is always convenient not only in terms of PLR but also in terms of throughput and packet delay. But considering stability to be one of our constraints, we can see that for $p_r = 1$ SA apparently offers the most reliable performance since after the throughput peak any CRDSA curve rapidly degrades with $N_B$ while SA degrades in a slow-paced manner. As a result, for moderate channel load lines having $G_T \leq 0.4$ for $N_B = 0$, SA is able to support up to approximately 450 users ensuring stability, while for CRDSA with 2 replicas the admissible population is approximately 50% less and for CRDSA with 3 replicas the admissible $M$ is less than 150. However, as previously said, the operational point for CRDSA always occurs in a point with higher throughput, smaller backlogged size and smaller packet delay. Moreover in Section V it has been shown that depending on the channel load line the actual $FET$ could be so big, that instability could be accepted. For example, for $M = 450$ and $p_0 = 0.089$ the computed $FET$ is so big that after a long simulation (up to $f = 10^{10}$) the cumulative distribution function is still equal to zero (thus we can assume the event of exit from the stability region really rare). Considering $M = 250$ and $p_0 = 0.2$ SA presents a globally stable behaviour around its throughput peak ($G_{OUT} = 0.36$) for $N_B^G = 65$ while CRDSA with 3 replicas presents a locally stable equilibrium point with throughput equal to 0.5 , $N_B^L$ close to 0 and unstable equilibrium point for $N_B^U \approx 43$. Using formulas in Section V the obtained average $FET$ is approximately $10^4$ frames. In this case, if we are willing to maintain the advantage of CRDSA with 3 replicas some countermeasures are required. In the next Section we will show simple yet effective control policies able to counteract the event of drift to saturation. After that, the convenience of using such a dynamic policy in the case of finite user population will be discussed.

### VIII. Control Limit Policies

Previous sections highlighted that statistical variations can yield to instability even in case of initially well-performing channels. To render a channel of this type stable 2 straightforward solutions can be used: the first one is to use a smaller value for the retransmission probability giving then rise to a larger backlogged population for the same throughput value; the second is to allow a smaller user population size $M$ thus resulting in a waste of capacity. A third solution is the use of control limit policies to control unstable channels by applying the countermeasures above in a dynamic manner. These dynamic control policies can be applied when it is not possible or it is not convenient to have a channel that is nominally stable. In this section we analyze control procedures of the limit type [11] able to ensure stability in the case of stationary parameters. However the same reasoning can also be applied in the case of dynamic ones (such as non-stationary $M$) with the necessary modifications. Two simple yet effective dynamic control procedures are considered: the input control procedure (ICP) and the retransmission control procedure (RCP). These two control procedures are based upon a subclass of policies known as control limit policies, in which the space of the policies is generally composed of two actions plus a critical state that determines the switch between them, known as control limit. In this case the control limit is a critical threshold for a certain number of backlogged users $\hat{N}_B$.

#### A. Input Control Procedure

This control procedure deals with new packets to transmit. In particular, two possible actions are possible: accept (action $a$) or deny (action $d$) and the switch between them is determined by the threshold $\hat{N}_B$ as previously mentioned.

#### B. Retransmission Control Procedure

As the name says, the retransmission control procedure deals with packets to retransmit and in particular with their retransmission probability. In particular two different retransmission probabilities $p_r$ and $p_c$ are defined that represent respectively the action taken in normal
retransmission state (action $r$) and in critical state (action $c$). From the definition above it is straightforward that it must be $p_r > p_c$. The switch between these two modes is determined once again by the threshold $\hat{N}_B$.

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{figure14}
\caption{Control limit policy examples.}
\end{figure}

Figure 14 graphically represents these 2 cases. As we can see both policies accomplish the same task of ensuring channel stability. However, while ICP controls the access of thinking users, RCP controls the access of backlogged users.

**IX. On the Use of Dynamic Control Policies for Finite Population**

In this section we discuss the result of the application of dynamic control policies for CRDSA with 2 and 3 replicas compared to SA and to the case of static design where $p_r = p_c$ is assumed. Consider the case previously analyzed in Section VII and reported in Figure 15 with $M = 250$, $p_0 = 0.2$ and $p_r = 1$. Moreover we assume $p_c = 0.75$ for CRDSA with 2 copies and $p_c = 0.53$ for CRDSA with 3 copies that are the biggest values for $p_c$ that ensure stability in the two respective cases.

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{figure15}
\caption{Equilibrium contour for SA and CRDSA with 2 and 3 replicas in normal and critical RCP state}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{figure16}
\caption{Throughput over $\hat{N}_B$ for dynamic retransmission policies}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{figure17}
\caption{Packet delay over $\hat{N}_B$ for dynamic retransmission policies}
\end{figure}

Figures 16 and 17 show the results in terms of throughput and average packet delay when varying the threshold of the control limit policy $\hat{N}_B$. As we can see the results among the various cases are similar in terms of maximum throughput and minimum delay achieved, except for ICP with 3 replicas that shows slightly higher throughput and lower delay than the other cases. For SA instead, the resulting channel for $M = 250$ and $p_0 = 0.2$ does not need any dynamic control policy since it is globally stable with throughput equal to 0.309 and delay equal to 1.76 frames. It is now clear that despite instability that needs to be controlled, the time spent in critical state is so small that results shown for CRDSA with 2 and 3 replicas are better than SA. Therefore we can affirm that CRDSA always gets better performance than SA despite instability.

It can also be noticed that in any case considered here, there is a certain range of flatness around $\hat{N}_B$. This range changes depending on the policy applied, the number of replicas, the time spent in critical state and the performance degradation between normal and critical state. The width of
this flat region is really important for the following reason: imagine a real application scenario with large delay (as it is the case in satellite communications). Differently from the case of immediate feedback assumed so far throughout the paper, when the $\hat{N}_B$ threshold is crossed the control limit policy needs a certain propagation time from the NCC to the satellite terminals before it is applied. As a result the effective threshold is not $\hat{N}_B$ but some value $\hat{N}_B + \delta$, where $\delta$ depends on the drift of the number of backlogged users for the particular case and on the propagation delay. Therefore the wider is the flat region, the less the performance optimality of the controlled channel will be influenced by the propagation delay [16].

Moreover, we can see that in the case of 2 replicas a degradation of the throughput is also found for $\hat{N}_B < N_B^P$. In general terms, when a control limit policy is applied it does not make sense to choose $\hat{N}_B < N_B^P$, therefore this result does not seem to be of interest for our analysis. However, when $N_B < \hat{N}_B$ the control limit policy switches back to normal state, but as previously mentioned users do not immediately receive the acknowledgment for this new change due to the propagation delay so that the critical policy might actually be left for some value $\hat{N}_B - \delta$ resulting in the possibility of suboptimal performance (see ICP with 2 copies per packet).

Let us now discuss the results when instead of dynamic control limit policy, the channel is designed by statically decreasing the retransmission probability so that the channel is always stable. Differently from SA, in this case decrementing $p_r$ has really small influence on the overall performance especially when the first point of intersection between the equilibrium contour and the channel load line takes place before the throughput peak and sufficiently far from it. As a matter of fact with $p_r = 0.53$ for CRDSA with 2 replicas and with $p_r = 0.75$ for CRDSA with 3 replicas the following results are obtained:

- CRDSA with 2 replicas: $G_{OUT} = 0.49; \text{Av}[D_{pkt}] = 1.01$
- CRDSA with 3 replicas: $G_{OUT} = 0.5; \text{Av}[D_{pkt}] = 1$

In conclusion we can say that with stationary values of $M$ and $p_0$, in most of the cases the use of dynamic policies for CRDSA does not appear to be a convenient solution compared to a static design. This can be affirmed also in light of the more realistic case of non-immediate feedback in which a certain performance degradation is present. The only case in which dynamic policies could be really necessary with stationary $M$ and $p_0$ is when $M$ is so big that the required $p_r$ rendering the channel stable really degrades the performance thus requiring the use of dynamic policies. Nevertheless the analysis above is useful for future work on the case of non-stationary $p_0$ and $M$ that does not allow to a-priori set the retransmission probability in order to be sure that the channel is always stable. Moreover the analysis presented for the case of finite $M$ is introductory to understand the application of dynamic policies to the case of infinite population.

X. On the use of dynamic control policies for infinite population

Finally we discuss the use of dynamic control policies in order to provide stability in the case of infinite population.
the bigger the $\lambda$ value, the higher the throughput but also the narrower the flat region. The reason for such a result is similar to what has been explained in the previous sections for the finite population case. In particular we can see from Table II that the closer the operational point is to the throughput peak, the smaller is the value of $N_B^U$. Figure 19 shows that obtained results concerning delay reflect throughput results since the wider is the flat region for the throughput, the wider is the flat region with delay equal to 1. Finally Figure 20 shows that for bigger values of $\lambda$ users spend some time in critical state also in the flat throughput region giving place on one hand to the possibility of suboptimality due to the propagation delay and on the other hand increasing the overall PLR by reason of rejected packets.

XI. CONCLUSIONS

In this paper a model for computation of the channel stability when using Contention Resolution Diversity Slotted Aloha as Random Access mode in DVB-RCS2 has been outlined. This model allows to study and predict stability and to calculate the expected throughput as well as the expected delay at the channel operating point. This kind of representation allows to have a simple yet effective view of how the communication behaves when changing one or more of the key parameters thus allowing a design that can easily consider the tradeoff among multiple constraints. A simplified framework for the calculation of the First Exit Time has also been introduced and its importance for the evaluation of unstable channels has been highlighted. All the presented models and tools can be extended to consider a different frame size, power balance, the introduction of First Error Correction or any other modification. However, concerning the frame size it is known from the literature that it does not bring any considerable advantage for sizes greater than 100-200 slots while the packet delay would be considerably increased. The central part of the paper discusses design settings and the convenience of using CRDSA over SA. In the last part, dynamic retransmission policies of the control limit type have been introduced and their effectiveness and convenience has been discussed against static design. In particular, concerning the case of finite user population, found results have shown that differently from SA in which dynamic retransmission policies are generally preferred, in the case of CRDSA a static design that ensures stability (i.e. by decreasing the retransmission probability for backlogged packets) is generally more convenient. Finally, dynamic retransmission policies have been applied to the case of infinite user population. Found results have shown that control limit policies are able to stabilize the channel also in the case of infinite user population while ensuring a performance close to the best achievable even in presence of large propagation delay as it is the case in a geostationary satellite system used for DVB-RCS communications.

REFERENCES
