UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

MASSIVE TDMA FOR SATELLITE COMMUNICATIONS

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Ponç Palau Puigdevall

In partial fulfilment

of the requirements for the degree in

Science and Telecommunication Technologies Engineering

Advisor: Josep Sala Álvarez

Barcelona, June 2017

Abstract

Nowadays, satellite communications are widely used to provide many services such as DVB – RCS (Digital Video Broadcasting Return Channel via Satellite) or IPoS (IP over Satellite). Among others, these two standards involve simultaneous connection of different gateways to the same satellite, transmitting small packets through a Random-Access technique. This requires a wise management of the Up-Link.

Although many RA techniques have been proposed, not all of them are suitable for satellite communications, for example the family of CSMA (Carrier Sense Multiple Access) techniques are not suitable for satellite networks because of the large channel propagation (about 250 ms for a GEO satellite). This fact leads us to think in other random access techniques to apply in a satellite network, such as TDMA (Time Division Multiple Access).

A lot of TDMA based RA schemes have been proposed such as Aloha, Slotted Aloha, Diversity Slotted Aloha and so on. This project aims to study a RA protocol called CRDSA (Contention Resolution Diversity Slotted Aloha) by simulating a MATLAB environment which recreates a satellite Up-Link using CRDSA.

Resum

Avui dia, les comunicacions via satèl·lit són àmpliament utilitzades a diferents aplicacions i serveis com per exemple el DVB – RCS (Digital Video Broadcasting Return Channel via Satellite) o el IPoS (IP over Satellite). Entre d'altres, aquests estàndards involucren comunicació simultània a un mateix satèl·lit. Aquest fet requereix una gestió intel·ligent en l'Up-Link.

Encara que moltes tècniques d'accés aleatori han estat proposades, no totes són adequades per a l'entorn satèl·lit. Per exemple, el protocol CSMA (Carrier Sense Multiple Access) no és adequat per a comunicacions via satèl·lit degut al llarg retard de propagació (250 ms per a un satèl·lit Geo-Estacionari). Aquest ens porta a pensar en altres tècniques d'accés a un medi compartit, com per exemple les tècniques basades en un multiplexat en el domini temporal, TDMA (Time Division Multiple Access).

Moltes tècniques basades en TDMA han estat proposades, són coneguts els protocols Aloha, el protocol Diversity Slotted Aloha, etc. Aquest projecte té com a objectiu estudiar un protocol d'accés aleatori anomenat CRDSA (Contention Resolution Diversity Slotted Aloha) mitjançant simulacions d'un Up-Link que utilitza CRDSA mitjançant simulacions amb el software MATLAB.

Resumen

Hoy en día, las comunicaciones vía Satélite son ampliamente utilizadas en diferentes aplicaciones y Servicios como por ejemplo el DVB – RCS (Digital Video Broadcasting Return Channel via Satellite) o el IPoS (IP over Satellite). Entre otros, estos estándares involucran comunicación simultánea a un mismo Satélite. Este hecho requiere una gestión inteligente en el Up-Link.

Aunque una cantidad considerable de técnicas ha sido propuesta, no todas las técnicas son adecuadas para un entorno Satélite. Por ejemplo, el protocolo CSMA (Carrier Sense Multiple Access) no es adecuado para comunicaciones satélite debido al largo tiempo de propagación (250 ms para un satélite Geo-Estacionario). Este hecho nos hace pensar en otras técnicas de acceso a un medio compartido, como por ejemplo las técnicas basadas en un multiplexado en el dominio temporal o TDMA en inglés (Time Division Multiple Access).

Muchas técnicas basadas en TDMA han sido propuestas, son conocidos los protocolos ALOHA, Diversity Slotted Aloha, etc. Este proyecto tiene como objetivo estudiar un protocolo de acceso aleatorio a un medio compartido llamado CRDSA (Contention Resolution Slotted Aloha) mediante simulaciones en MATLAB de un entorno satélite en el que el Up-Link utiliza el protocolo CRDSA.

Revision history and approval record

Revision	Date	Purpose
0	01/05/2017	Document creation
1	22/06/2017	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Ponç Palau Puigdevall	ppalaupuigdevall@gmail.com
Josep Sala Álvarez	Josep.sala@upc.edu

Written by: Ponç Palau Puigdevall		Reviewed and approved by: Josep Sala Álvarez	
Date	19/06/2017	Date	22/06/2017
Name	Ponç Palau Puigdevall	Name	Josep Sala Álvarez
Position	Project Author	Position	Project Supervisor

Table of contents

Abstract 1

Resum 2

Resumen 3

Revision history and approval record 4

Table of contents 5…6

List of Figures 7

List of Acronyms 8

1. Introduction 9…10
   - 1.1. Project Outline 9
   - 1.2. Organization 9
   - 1.3. Contributions 10
2. Proposed Random Access Scheme 11...12
   - 2.1. System Assumptions 11
   - 2.2. CRDSA Description 12
3. System Structure 13…29
   - 3.1. Overall System Structure 13
   - 3.2. Transmitter 14…19
     - 3.2.1. Packet Structure 14…19
       - 3.2.1.1. Preamble Symbols 14
       - 3.2.1.2. Payload Symbols 16
       - 3.2.1.3. Guard Symbols 18
     - 3.2.2. Pulse Shaping 19
   - 3.3. Channel 20…22
   - 3.4. Receiver 22…29
     - 3.4.1. Packet Detector & Packet Decoding 22
     - 3.4.2. Successive Interference Cancellation Algorithm 26
4. Results 30…35
   - 4.1. Ideal – Estimation Case 31
   - 4.2. Non – Ideal Estimation Case 33

5. Budget 36

6. Conclusions and Future Development 37

Bibliography 38

List of Figures

Figure 0. Gantt Diagram

Figure 1. CRDSA Scenario

Figure 2. Frame Structure & CRDSA Example

Figure 3. MISO System

Figure 4. Auto correlation and Cross Correlation for Npre = 31

Figure 5. Auto correlation and Cross Correlation for Npre = 63

Figure 6. Payload Bits Raw Structure

Figure 7. Payload Bits CRC Structure

Figure 8. Packet Generator Block Diagram

Figure 9. Pulse Shaping Block Diagram

Figure 10. Attenuation for different users' locations

Figure 11. Geometric attenuation for K=1000 users

Figure 12. Packet Detection Peak

Figure 13. Packet Detection & Packet Estimation Block Diagram

Figure 14. Packet Decoding Block Diagram

Figure 15. Receiver Example

Figure 16. Receiver Example Decoded Packets

Figure 17. Receiver Example Updated Packets

Figure 18. Receiver's State Diagram

Figure 19. CRDSA Throughput Curves for Ideal Estimation Case

Figure 20. CRDSA PER curves for Ideal Channel Estimation

Figure 21. CRDSA Throughput Curves for Non-Ideal Estimation Case

Figure 22. CRDSA PER curves for Non-ideal Estimation Case

Figure 23. Throughput Comparison Ideal against Non-ideal Niter=3

Figure 24. Throughput Comparison Ideal against Non-ideal Niter=4

List of Acronyms:

ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
CRC	Cyclic Redundancy Check
CRDSA	Contention Resolution Diversity Slotted Aloha
DSA	Diversity Slotted Aloha
DVB-RCS	Digital Video Broadcast – Return Channel Over Satellite
FEC	Forward Error Correction
MF	Multi-Frequency
MISO	Multiple Input Single Output
QPSK	Quadrature Phase Shift Keying
RA	Random Access
SIC	Successive Interference Cancellation
SIR	Signal to Interference Ratio
TDMA	Time Division Multiple Access

1. Introduction

1.1. Project Outline

a. Objectives

The objective of this project is to develop a MATLAB® environment which recreates a satellite Up-Link scenario which uses CRDSA RA technique.

b. Requirements and Specifications

The environment must simulate situations in which many users can be operative. The number of users has to be massive. Hence, the software must be programmed taking that into account.

The software is created entirely from the beginning of the project, no previous software is used.

1.2. Organization

The organization of the project has been the indicated in the following Gantt Diagram:

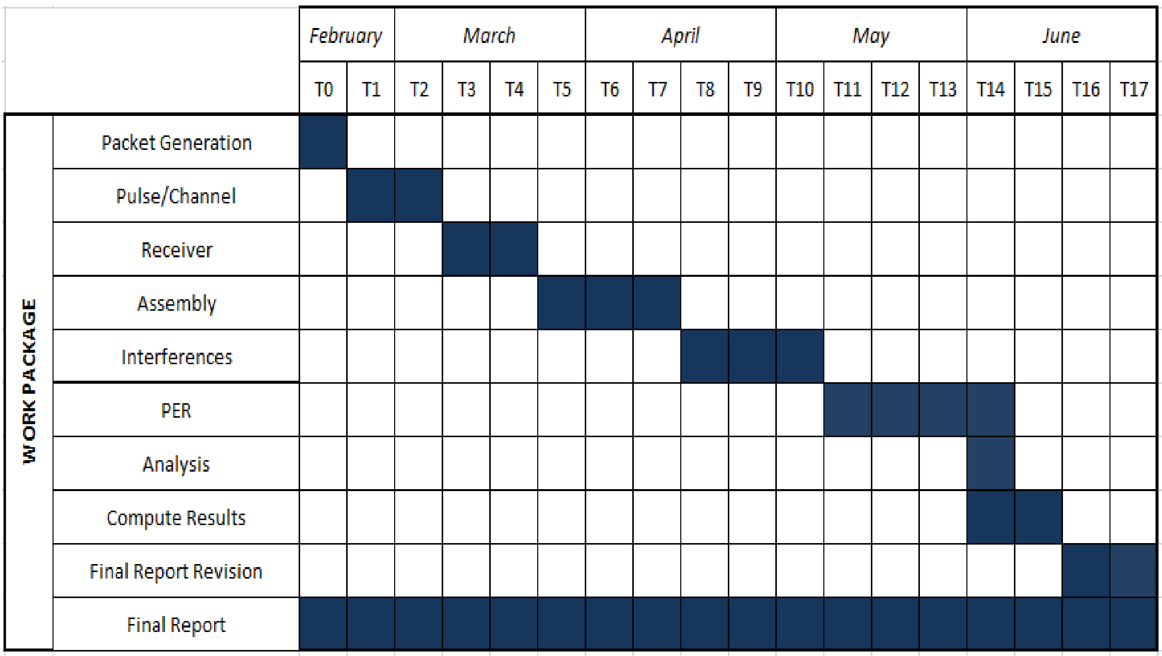

Figure 0. Gantt Diagram

Regarding the meetings with the project's supervisor, these were not strictly agreed. Every 2 weeks a supervising meeting was held in order to check the project execution.

1.3. Contributions

This project is based on a technique proposed in [1]. This technique proposes a random access technique which is based in a time division multiple access scheme. This technique was proposed to me by the project supervisor.

2. Proposed Random Access Scheme

Contention Resolution Diversity Slotted Aloha (CRDSA) is a TDMA based random access technique which increments the capacity of the network by using a Successive Interference Cancellation (SIC) algorithm. Although its main use is in Satellite Networks, it is not only restricted to this environment. The Up-Link of a satellite network will be examined, where the satellite is shared between the different gateways. This scenario represents a MISO (Multiple Input Single Output) case. The next picture shows the studied scenario.

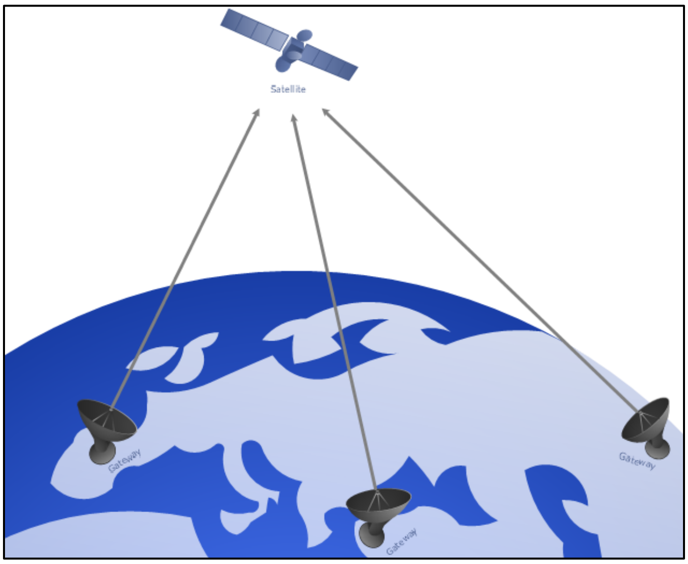

Fig. 1. CRDSA Scenario

The CRDSA scheme is an improved version of Diversity Slotted Aloha and Slotted Aloha (DSA) itself. As DSA does, the CRDSA protocol uses many replicas in the same frame at a random time selected by the transmitter. In this project only 1 replica has been considered. However, the programmed software allows setting more than 1 replica. While DSA does not do anything but detect and decode packets in reception, CRDSA includes an iterative Cancellation Algorithm which uses the information of the successful decoded packets to handle with packet collisions that occur with a certain probability in a given slot.

The main CRDSA advantage lie in the improved packet loss ratio and the higher operational throughput compared to SA and DSA.

2.1 System Assumptions

Although some standards such as DVB-RCS use Multi Frequency TDMA (MF-TDMA), this project will be focussed on one single carrier per transmitter for a reason of simplicity.

It is assumed that all gateways are already synchronized to the network, this decision has been taken because the network synchronization process is out of the scope of this project. However, a guard time has been left at the end of each packet to recreate a real version of the packets transmitted.

It is also assumed that all the users in the network transmit at same nominal power $P_{TX}[i] = P_{TX}$ and same symbol rate $Rs[i] = R_S = \frac{1}{T_S}$. In order to generate power diversity at reception, different channel attenuations have been applied.

2.2 CRDSA Description

The proposed TDMA frame structure is drawn in the figure below. Each RA frame is composed of M_slots. Each terminal transmits two packets or *bursts* along the frame to enhance the probability of packet success (the packet is successfully decoded). In fact, these two packets are identical. As it is detailed later on this report in section 3.2.1, each packet contains signalling information about which slot the twin packet will be located in. The receiver will use this recovered signalling information to cancel the twin burst that may be interfering in the twin slot. The next example shows a realization where the algorithm used by CRDSA is explained.

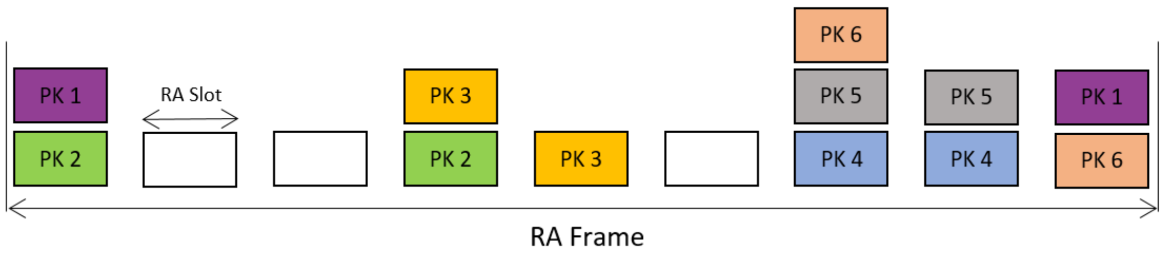

Fig. 2. Frame Structure & CRDSA Example

In the example provided in Fig. 2. the packet 1 cannot be recovered because a collision occurred between packet 1 and packet 2 in slot 1. Then the receiver goes to slot 4 and cannot recover packet 2 and packet 3 due to a collision. The receiver goes to slot 5 and decodes successfully packet 3. The receiver uses the signalling information in packet 3 to regenerate this packet and to cancel it in slot 4. Now, due to the cancellation, packet 2 is alone in slot 4 and can be recovered successfully. The receiver cancels packet 2 in slot 1 and recovers packet 1 in slot 1. Then the receiver moves to slot 9 where a collision between packet 1 and 6 occurred. The receiver cancels packet 1 and recovers packet number 6. After that, it cancels packet number 6 in slot 7. Then the receiver tries to decode packet 5 or packet 4 in slot 7 but it fails because they are mutually interfering in both slot 7 and slot 8. The situation in slot 7 and 8 will be called a “loop” because two packets mutually interfere and the receiver doesn’t have information about any of them from other slots.

3. System Structure

The main objective of this project is to develop a Matlab® environment which simulates the scenario proposed in fig. 1. using CRDSA method for random access. In this section the designed and implemented (programmed) structure is given, as well as the considerations taken into account at the time of developing the environment.

3.1 Overall System Structure

In the simulations done in this project it has been assumed that K is the number of global users in the satellite network, that is, all the users that are registered to that satellite network but not necessarily active all the time. These users can be active in a certain slot with a certain probability p, which can be set at each simulation. So, the average number of users transmitting in a certain slot is:

$$K \cdot p = \text{Average number of users in each slot}$$

The number of global users K is assumed to be high, around magnitudes of 1000 or 10000. These high K values are the reason why the term “Massive” has been used in the project title.

The system’s load G is defined in packets per slot (packets/slot), the definition of this load allows performing iterations around a certain number of packets/slot. Thus, the load as a function of the slot is given by:

$$G\left[\frac{packets}{slot}\right] = K(slot) \cdot p$$

The overall system’s block design can be drawn as follows, it must be taken into account that there are $K \cdot p$ possible transmitters in each slot.

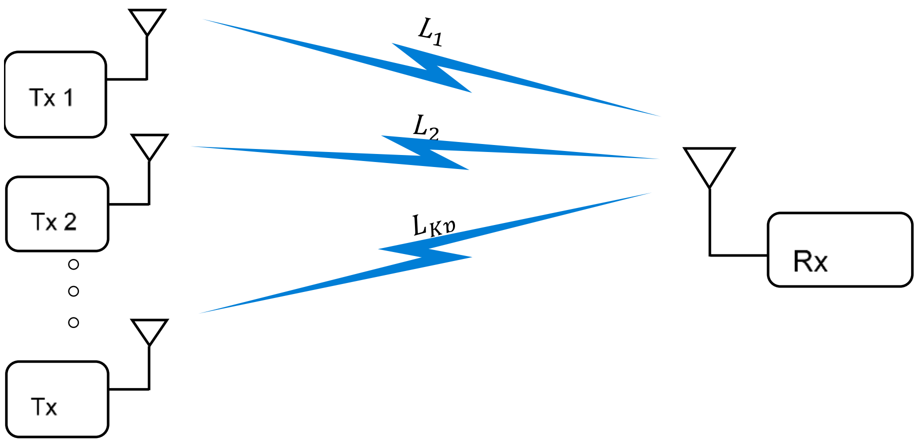

Fig.3. MISO System

Each block of the previous image is explained in the following sections.

3.2 Transmitter

All users registered in the satellite network transmit with the same packet length and structure. According to CRDSA protocol, each user transmits the packet twice and it must be indicated in which slot the twin packet will be transmitted.

The receiver must check if a packet is received correctly and to be able to do that, the transmitter must introduce a Cyclic Redundancy Check (CRC) that the receiver will check to determine if a packet is successfully decoded or not.

Moreover, satellite communications must be robust to the errors introduced by the channel and there are different solutions to do so. The first one is Automatic Repeat Request (ARQ) and the second one is Forward Error Correction (FEC).

ARQ is discarded when designing a satellite link due to the large propagation that the channel introduces (250 ms for a GEO satellite) and asking for a re-transmission is nowhere near efficient. Therefore, FEC is a good option when designing a satellite link because it is capable of correcting errors on the receiver side, adding no delay to the communication with respect to ARQ.

The different parts of the transmitter block are explained in the next sub-sections and a final draw of the block is given at the end of the transmitter section.

3.2.1 Packet Structure

3.2.1.1 Preamble Symbols

TDMA networks require a synchronization process to receive the user's bursts at the beginning of a certain slot. This is achieved by exploiting the burst preamble. The purpose of this preamble are both detect a packet in a certain slot and perform a channel estimation. This channel estimation is used by the receiver to cancel the interference caused by the decoded packet in its twin slot. These steps commented will be explained further in this project because of their relevance inside this project.

The pseudo-random preamble sequences are known by the transmitter and the receiver. The transmitter selects randomly a preamble sequence among the family of preamble sequences of size S_{PR}. The transmitter will use the same preamble sequence for the twin packet. The family of preamble sequences of length N_{pre} shall provide good auto and cross-correlation properties. It has to be reminded that the main purpose of the preamble sequence is to provide a good channel estimation and to detect a packet in a certain slot.

The preamble sequences used in this project are the families of pseudo-random sequences of Gold Codes. Gold codes provide the necessary specifications (good and cross-correlation) to accomplish the two main functions of the preamble. The families chosen have a length of $N_{pre} = 31$ and $N_{pre} = 63$.

To generate these groups of sequences four polynomials have been used. The next table summarizes the polynomials used to create each family of sequences.

	$N_{pre} = 31$	$N_{pre} = 63$
$Gp_1(D)$	$D^5 + D^2 + 1$	$D^6 + D + 1$
$Gp_2(D)$	$D^5 + D^4 + D^3 + D^2 + 1$	$D^6 + D^5 + D^2 + D + 1$

The next figures show the autocorrelation and cross-correlation with some sequences of different preamble sequences for cases $N_{pre} = 31$ and $N_{pre} = 63$.

Before analysing the images, let us introduce the notation of them. The correlation between sequences have been defined as:

$$C(k) = \sum_{n=0}^{N_{pre}-1} p_x(n) \cdot p_y(n-k)$$

Where $p_x(n)$ is the sequence number x among all N_{pre} sequences.

In the figures below the correlations are show for different values of N_{pre}.

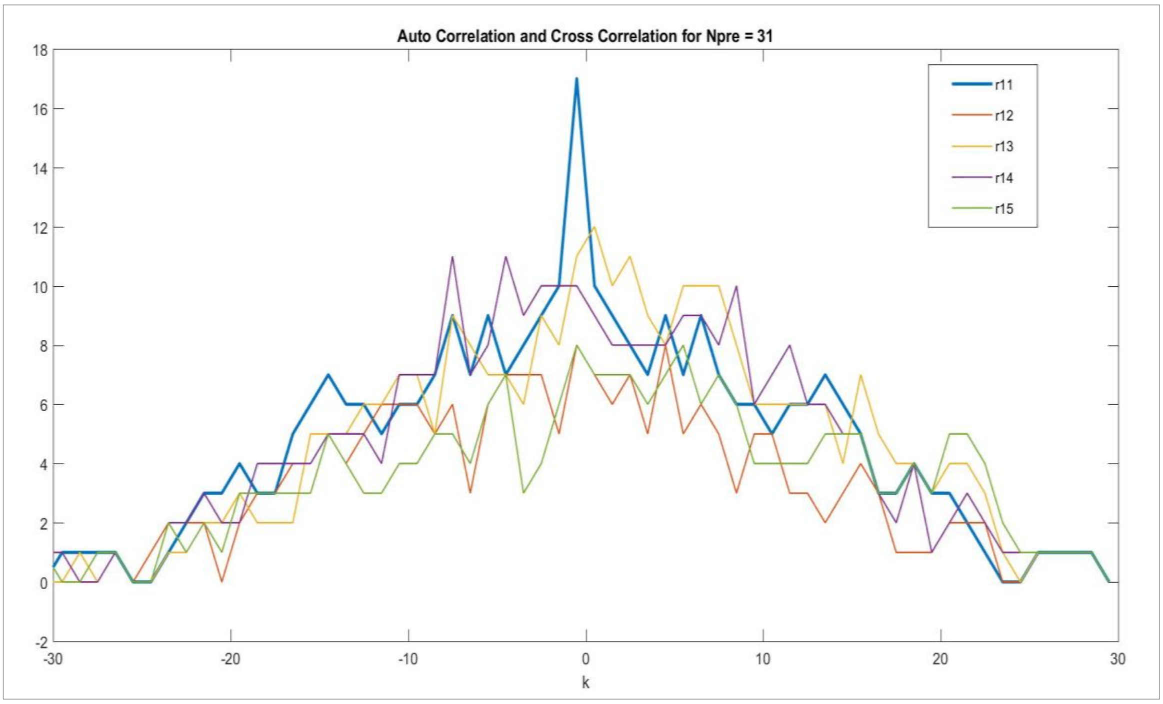

Figure 4. Auto correlation and Cross Correlation for Npre = 31

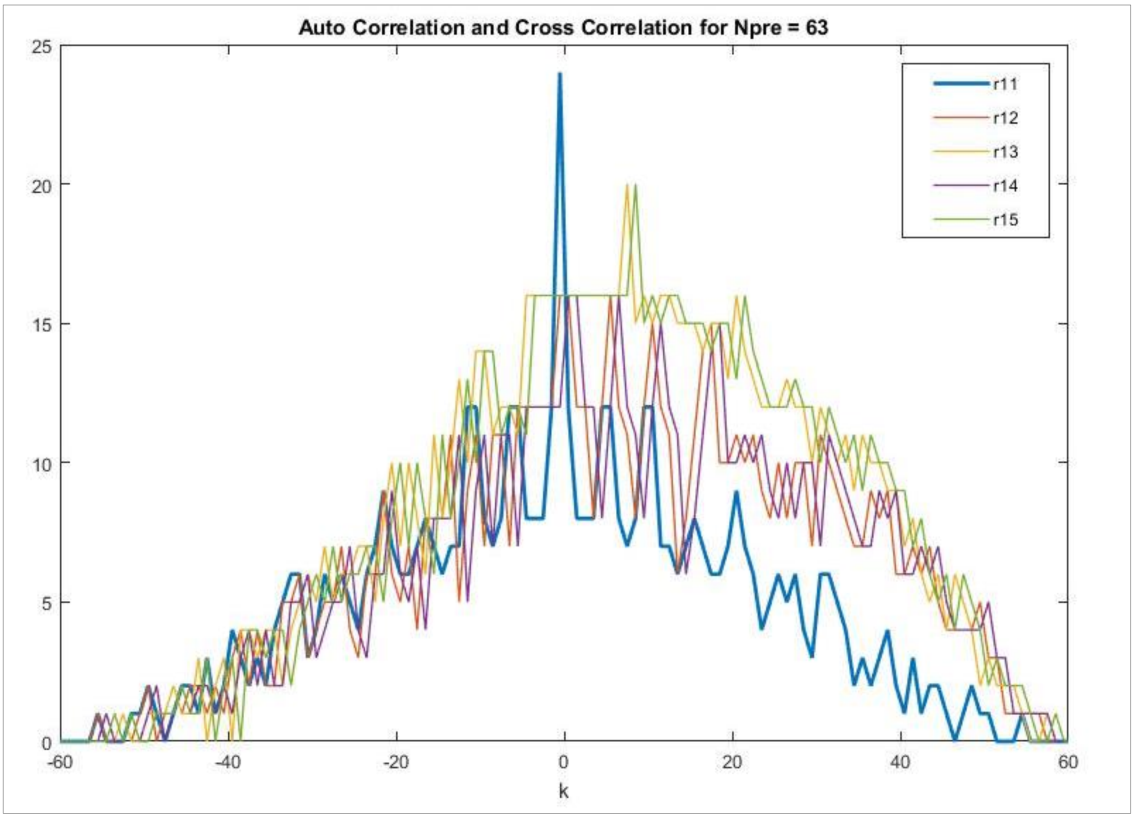

Figure 5. Auto correlation and Cross Correlation for Npre = 63

As can be appreciated in the figures above, the longer the length is, the larger the peak of the autocorrelation. This fact will lead to a better channel estimation (channel estimation process will be explained in section 3.4.1).

The packet binary sequences will be mapped using BPSK modulation. These BPSK modulated symbols will be represented as:

$$\underline{s}_{pre}[i] = [c_1[i], c_2[i], \ldots\ldots, c_{N_{pre}}[i]]$$

Where $c_x[i] = \{-1, +1\}$

3.2.1.2 Payload Symbols

The payload is the most important part of the packets, it contains data, user's controlling information, and signalling information to indicate where the twin packet is located within a frame. Therefore, it must be properly protected against channel errors. Not only that, the payload bits must include a Cyclic Redundancy Check (CRC) to allow the receiver deciding if the payload bits are correct or wrong.

To protect the payload bits, a Forward Error Correction technique has been used. Among all FEC techniques (Block Coding, Convolutional Codes, Turbo Codes and Low-Density Parity- Check Codes (LDPC)), the technique chosen has been a convolutional code.

Before explaining in detail how the payload has been encoded, let us detail the payload structure.

The payload has three main fields: the twin signalling information, the user number and the data. The next figure summarizes how many bits have been reserved for each field.

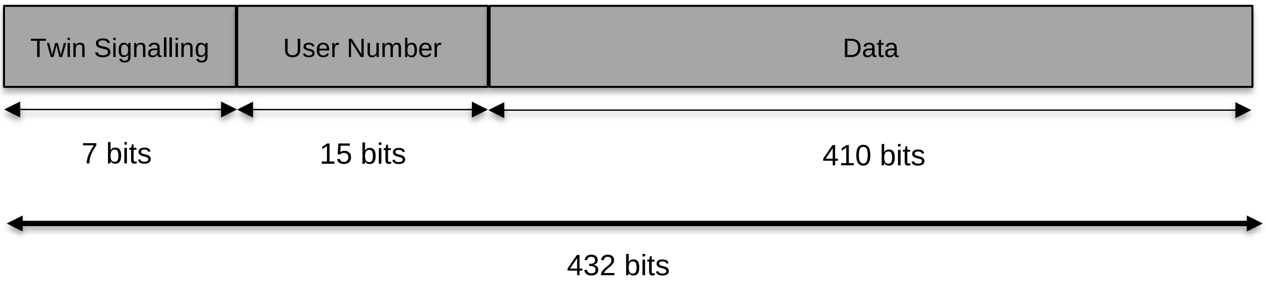

Figure 6. Payload Bits Raw Structure

With these amounts of bits, we can represent:

- A packet inside a frame of composed by $2^7 = 128$ slots (although in the simulations a frame of 100 slots has been considered).
- A total number of users up to $2^{15} = 32768$ users (although in the simulations the number of global users has been defined as K = 1000),

The CRC must be applied to this bit structure, what the CRC does is to divide the 432 bits in 4 sequences and for each of these 4 sequences add 3 bits of CRC. This is due to the degree of the generator polynomial. The one used for this CRC is $D^3 + 1$. The payload bit structure is shown in the following figure.

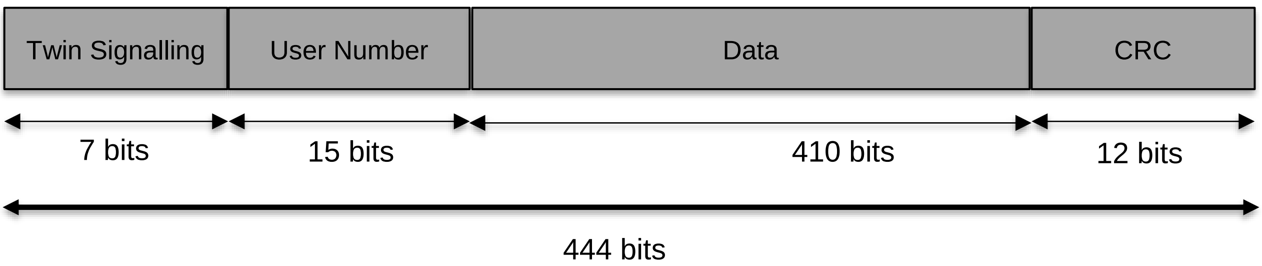

Figure 7. Payload Bits CRC Structure

Once the CRC is applied to the payload bits, the FEC encoding must be done. As it has been stated before in this project the proposed FEC technique has been a convolutional code with the following parameters:

$$g = [133_8, 171_8] = [1011011_2, 1111001_2] =$$
$$= [D^6 + D^4 + D^3 + D + 1, D^6 + D^5 + D^4 + D^3 + 1]$$

The rate of this convolutional code is $r = 1/2$.

So, when the convolutional code is applied the transmitter has $444 \cdot 2 = 888$ bits to transmit. These bits are QPSK modulated.

This fact leads to a total of $N_{paysim} = 444$ QPSK symbols. Therefore, the modulated payload symbols $\underline{s}_{pay}[i]$ (i refers to the user) is of can be expressed as:

$$\underline{s}_{pay}[i] = \frac{1}{\sqrt{2}}[d_{p,1}[i] + jd_{q,1}[i], \ldots\ldots, d_{p,N_{paysim}}[i] + jd_{q,N_{paysim}}[i]\,]$$

Where $d_{p,x}[i], d_{q,x}[i] = \{-1, +1\}$

3.2.1.3 Guard Symbols

Although in this project the system is assumed to be synchronized, it is not the case in real systems. Hence, a guard period of $5 \cdot T_S$ seconds has been considered. The guard symbols are expressed as:

$$\underline{s}_{guard}[i] = [0\ 0\ 0\ 0\ 0]$$

Once the three parts of a packet have been detailed, a block diagram of the packet is shown to summarize the packet generation process.

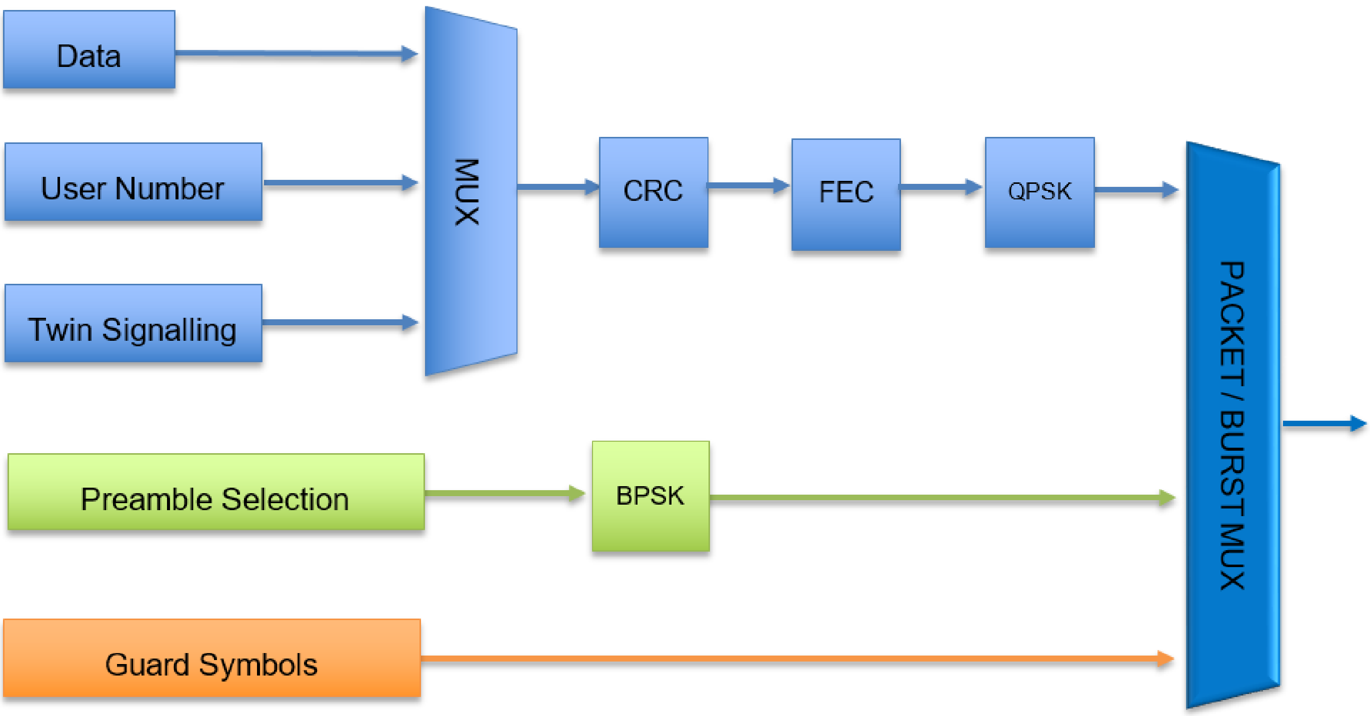

Figure 8. Packet Generator Block Diagram

At the output of the Packet Generator block we have the **complex** symbols that have to be transmitted. These symbols can be expressed as:

$$B[n] = [Preamble\ Symbols, Payload\ Symbols, Guard\ Symbols] = I[n] + jQ[n]$$

After creating the packet complex symbols, the transmitter must filter the data through the shaping pulse.

3.2.1 Pulse Shaping

The pulse that has been used to filter the complex symbols is a square root raised cosine with a roll-off factor of 0.25. The pulse's energy is equal to 1. As it is not necessary to develop the entire project, signal's modulation will not be considered and the signal will be always treated as a base band signal.

The signal transmitted by each user is defined as:

$$b_s(n) = \sqrt{P_{TX}} \cdot \big(i_s(n) + j\, q_s(n)\big) = \sqrt{P_{TX}}\ (\sum_{n=-\infty}^{+\infty} I[n] \cdot p(t-nT) + j \sum_{n=-\infty}^{+\infty} Q[n] \cdot p(t-nT)\ )$$

As has been stated before in this report, all the users transmit with the same power P_{TX} .

The block diagram to perform the pulse shaping is the following:

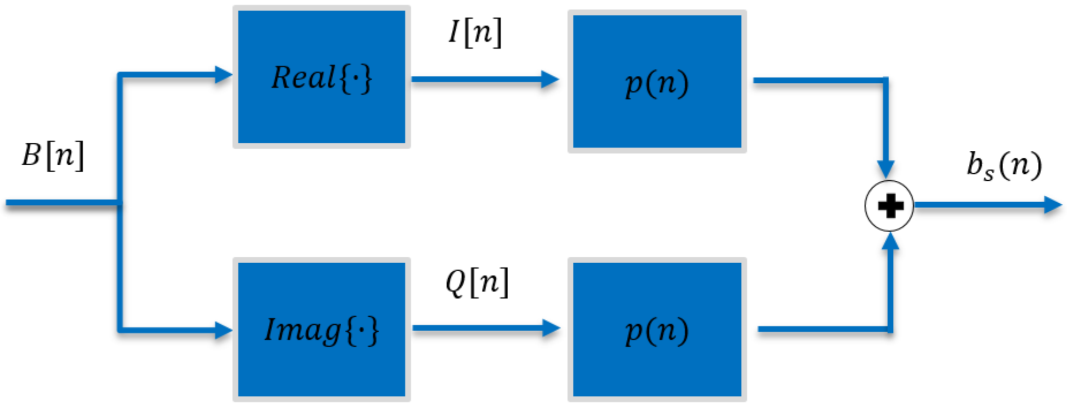

Figure 9. Pulse Shaping Block Diagram

3.3 Channel

The simulated channel is described in this section. The channel proposed for the satellite Up-Link scenario represents a situation in which the K users registered to the network are uniformly distributed in the satellite coverage area. This means that different regions will be considered for different locations inside the coverage area of the satellite. Concretely, circular regions inside the satellite coverage area have been defined as represented in the next image:

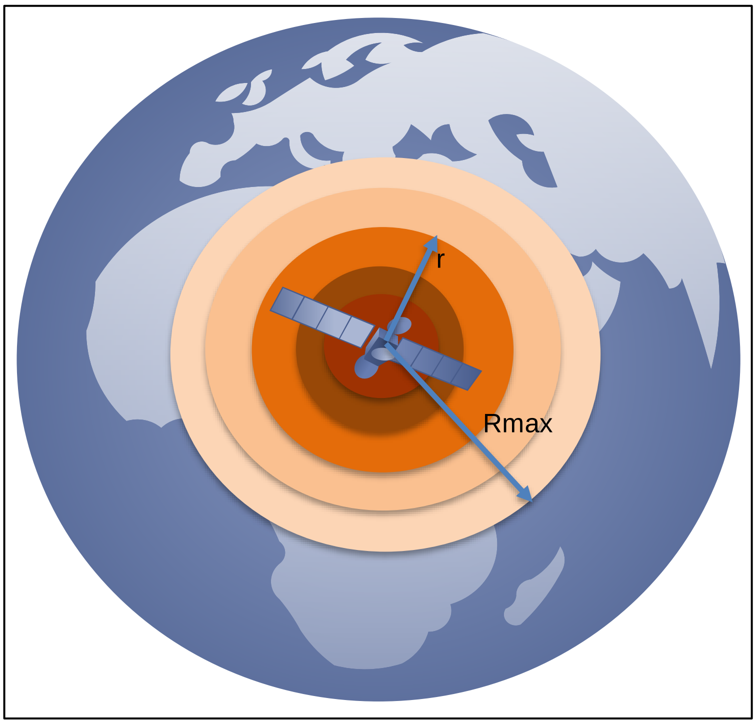

Figure 10. Attenuation for different users' locations

Each user will have a certain random attenuation that will be the same in one frame and consequently in the slots that the users sends the twin packets. This attenuation is given by the following expression:

$$h_0[i] = e^{-\frac{r[i]^2}{\sigma_r^2}}$$

This attenuation will be maximum when $r = R_{max}$ and will be minimum when $r = 0$, this is, the user is located under the satellite. This attenuation is representative and it has been considered to have a certain deviation in the power received by users in the same slot. The users in the areas with larger r (low attenuations) will be more likely to appear due to the size of the circumference of that given r. Only a few users will be located under the satellite, so the probability of having an attenuation equal to 1 is minor. In order to show that fact, a random realisation of r values has been done and the behaviour commented in the lines above can be appreciated.

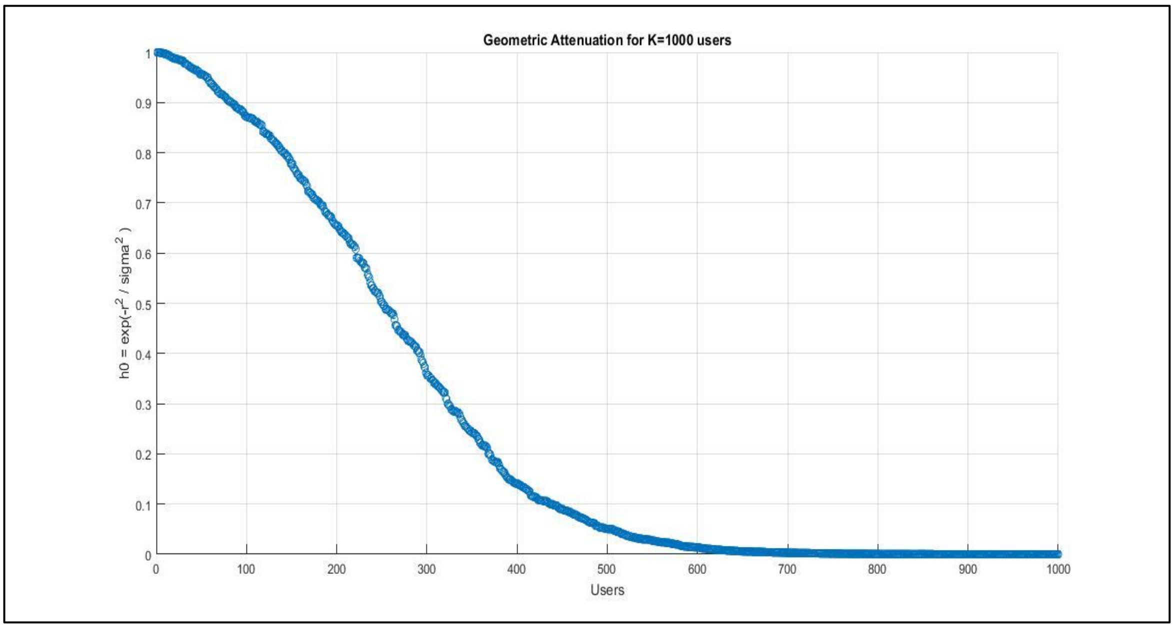

Figure 11. Geometric attenuation for K=1000 users

Not only does the channel introduce attenuation, it also contains a phase ($\theta[i]$). This phase is modelled as a uniformly distributed variable between $[0, 2\pi]$.

Having defined the attenuation and phase of the channel for a single user located at a certain distance r, we can give its complete expression.

$$h[i] = h_0[i] \cdot e^{+j\theta[i]} = e^{-\frac{r[i]^2}{\sigma_r^2}} \cdot e^{+j\theta[i]}$$

An important thing to mention is that CRDSA, as it will be stated later in this project, is not near-far resistant. This fact led me to define this kind of attenuation which distributes exponentially the power received by the different users in the same frame but permits having users with almost the same power in the same slot, which will help us to proof that the system is not near far resistant.

In addition to the channel propagation attenuation, a complex AWGN channel of power $E\left\{\left|\underline{n}\right|^2\right\} = N_0$ is added to the signal. This complex AWGN is defined as:

$$\underline{n} = \underline{n_I} + j \cdot \underline{n_Q} \text{ , where } E\left\{\left|\underline{n_I}\right|^2\right\} = E\left\{\left|\underline{n_Q}\right|^2\right\} = \frac{N_0}{2}$$

Therefore, the received signal in one slot for a single user will be:

$$r_x(n) = h[1] \cdot b_{s,1}(n) + n(n) \rightarrow \text{Expressed in vectorial terms} \rightarrow \underline{r_x} = h[1] \cdot \underline{b_{s,1}} + \underline{n}$$

Finally, the received signal for all users **active** in a given slot is, on average:

$$r_x(n) = \sum_{i=1}^{K \cdot p} h[i] \cdot b_{s,i}(n) + n(n)$$

3.4 Receiver

In this section, the receiver is accurately analysed. This is one of the most important parts of the project because the receiver is the part that manages the information of all users that transmit in the same slot. Before going into details about each receiver's part, we will briefly describe an overview of what does the receiver do.

The receiver will operate slot by slot. When the receiver analyses one slot, it will search for packets. Once it detects a packet, it will try to decode successfully that packet, this is, checking the CRC packet, if the CRC is checked successfully, the receiver marks that packet as a "clean" packet. Then, the receiver will move on to the next slot and will perform these already mentioned tasks again. When the receiver finishes a frame, it will perform a Successive Interference Cancellation algorithm that will be later explained.

3.4.1 Packet Detection & Packet Decoding

The main objective of this part of the receiver is both to detect a packet in each slot and to decode that packet. Firstly, the packet detection procedure will be analysed.

The packet detection process is achieved thanks to the preamble detector. What this detector does is to perform a parallel correlation calculus between the received signal and all the possible preamble sequences of size N_{pre} known by the receiver and by the transmitter. When the correlation maximum is above a certain threshold, the receiver detects a packet. When this happens, the receiver performs the channel estimation thanks to the preamble correlation process. The receiver aims to estimate the parameter $h[i]$ specified in section 3.3.

Before explaining in detail the channel estimation procedure, the threshold setting is explained. The threshold must be properly selected to detect a packet, if no packet is detected, the receiver will not even try to decode whatever it is in that slot. The threshold has been set high enough to not detecting a packet when there are no users in that slot (there is only AWGN in that slot) but low enough to detect users with a low $\frac{E_S}{N_0}$.

The final value was acquired through simulation, executing a lot of realisations of AWGN samples and correlating them with all the preambles. The maximum values of the correlations were computed and the threshold was set to the double of this maximum.

The next figure shows the high correlation peak obtained in the received signal of a certain slot.

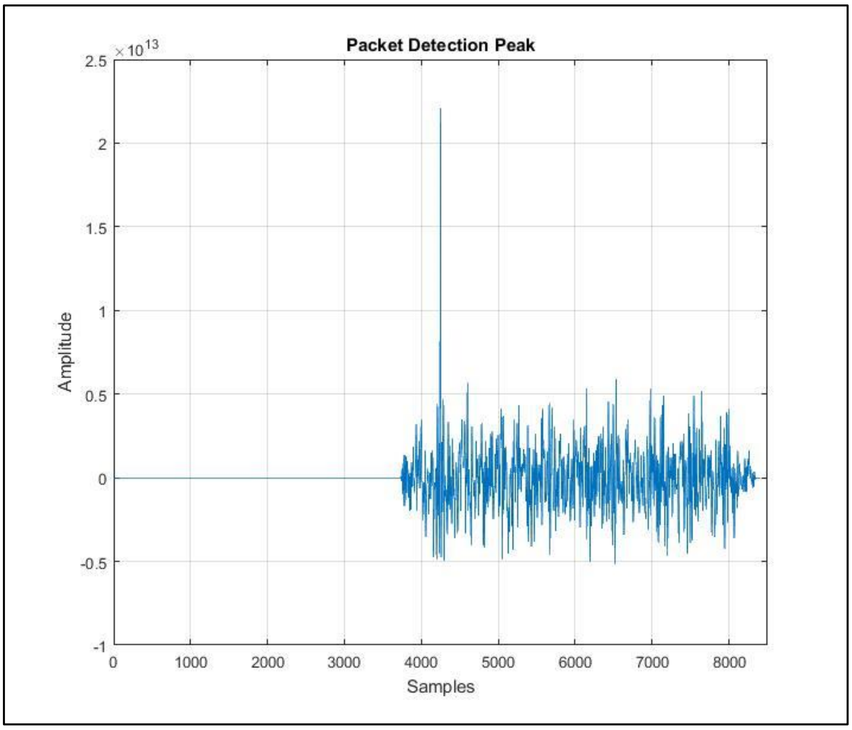

Figure 12. Packet Detection Peak

In the picture above the units in the Y axis may lead to a misunderstanding. These amplitude values are so high because in all the simulations the system is forced to be limited by interferences, not by noise. Due to this fact, we wanted all the users SNR to be almost infinity, being other users signal present in the same slot the unique source of interference.

Once the packet detection process is fully explained, the channel estimating process is analysed.

To clarify the channel estimation process, the following example of it is fully examined.

Supposing that we had the following signal at the input of the receiver:

$$s[n] = h(p[n] + d[n]) + w[n]$$

Where $p[n]$ represents the preamble sequence and $d[n]$ represents the data sent by the user under example. The parameter h represents the channel attenuation to be estimated.

What the receiver does is to correlate the received sequence $s[n]$ with all the possible preamble sequences in the family known by the transmitter and the receiver of size N_{pre}.

The correlation operation $C(\epsilon)$ is defined as it follows:

$$C(\epsilon) = \sum_{m=0}^{N_{pre}-1} s[m+\epsilon] \cdot p^{*}[m]$$

If we develop this expression we obtain:

$$C(\epsilon) = \sum_{m=0}^{N_{pre}-1} \left(\underbrace{h \cdot p[m+\epsilon] \cdot p^{*}[m]}_{\text{Desired part of the correlation}} + \underbrace{h \cdot d[m+\epsilon] \cdot p^{*}[m] + w[m+\epsilon] \cdot p^{*}[m]}_{\text{Interference part in terms of channel estimation}} \right)$$

This equation can be split in two parts, the part that is useful in terms of channel estimation and the part that interferes the useful part. If the length of the preamble sequence is long enough, then the previous expression can be expressed as:

$$C(\epsilon) \approx h_0 \cdot \sum_{m=0}^{N_{pre}-1} p[m+\epsilon] \cdot p^{*}[m]$$

The value that the receiver searches for is the value of ϵ that accomplish:

$$\underset{\epsilon}{\operatorname{argmax}} \{ \, C(\epsilon) \, \}$$

This is, the value of ϵ that maximizes the auto correlation of the pulse, if the signal is not delayed, this point will be $\epsilon = 0$. If we replace ϵ by 0 what we obtain is the Energy of the Preamble sequence (E_p). This ϵ value that accomplishes that is called ϵ_{opt}.

$$\epsilon_{opt} = 0 \rightarrow C(\epsilon = 0) = h \cdot E_p$$

To obtain the estimation of the channel amplitude and phase the receiver has to compute:

$$\hat{h} = \frac{C(\epsilon_{opt})}{E_p} = \frac{C(0)}{E_p} = |\widehat{h_0}| \cdot e^{j\hat{\theta}}$$

To finish the channel estimation procedure, the receiver compute both the modulus and the argument of this $\hat{h}$ value.

Once the example is finishes a block diagram is provided to summarize the packet detection and packet estimation procedures.

To detect a packet in a certain slot, the receiver performs the parallel preamble correlations as shown in the next picture:

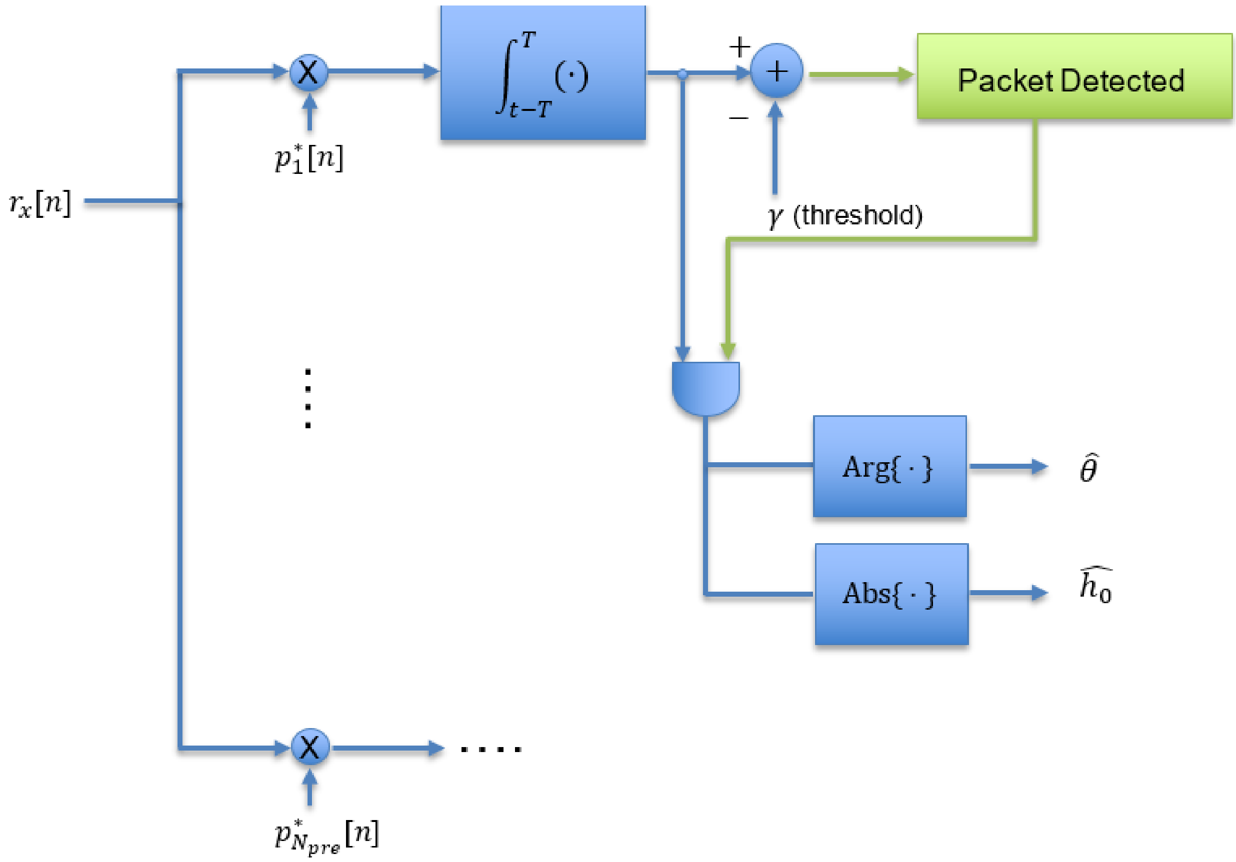

Figure 13. Packet Detection & Packet Estimation Block Diagram

When the receiver has performed the packet detection and the channel estimation, it must decode the packet detected. If no packet has been detected, then it moves to the next slot. The process to decode a packet is specified in the next paragraph, but before doing that, a block diagram is drawn to clarify what does the receiver does to decode a packet.

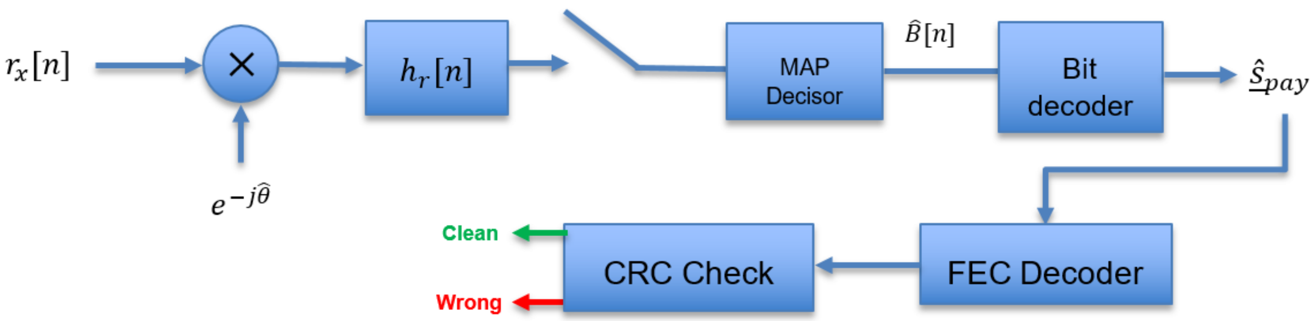

Figure 14. Packet Decoding Block Diagram

Some parameters of the above block diagram must be defined.

The matched filter is defined as $h_r[n] = p^*[T - n]$ .The FEC decoder is performs a soft decision.

Apart from that, the decoder has been programmed in a way that allows a second (or the number of times we set) attempt when decodes a packet. Having set a certain number of attempts, if the receiver does not decode successfully a packet in a given slot, it will perform the decoding process again in that slot instead of moving on to the next slot.

3.4.2 Successive Interference Cancellation Algorithm

On this section, the algorithm which cancels the interferences caused between users is studied. An overview of the algorithm has been explained in section 2.2. Hence, on this section the algorithm is accurately studied.

0. **Set receiver parameters**

What the receiver does at every single beginning of a frame is clear all its buffer. It also sets the number of iterations defined as N_{iter} to $Niter = 1$.

1. **Detection & Decoding of clean bursts**

In this step, the receiver decodes as “clean” packets those packets for which the signal, noise and interference level allow detecting the packet and decoding the packet’s payload successfully.

It’s in these steps where the receiver performs the steps explained in section 3.4.1 (Detection and Decoding).

When a burst is successfully decoded, the receiver regenerates that burst at baseband level with the payload bits that it has decoded. When it regenerates the burst at baseband level, it uses the amplitude estimated using the steps described in section 3.4.1. It doesn’t use the estimated phase because it cannot be considered equal in all the Frame duration. The regenerated baseband signal of the burst can be expressed as:

$$\hat{h}[i] \cdot \widehat{b_s}(n)$$

Not only does that, the receiver extracts the twin signalling information of the decoded payload bits. This field indicates where the twin burst will be transmitted. Using this information, the receiver stores the regenerated baseband signal in its buffer in order to cancel this signal as it will be explained in the next step.

2. **Contention Resolution Algorithm**

When the receiver analyses one entire frame, this is, analyses the 100 slots in that frame trying to detect & decode the packets, it performs the successive cancelation technique in the slots in which the users from the decoded packet have transmitted the replicas bursts.

To clarify this process of cancelling the interferent signals, an example is given:

Supposing a hypothetical situation with a frame composed by 4 slots and 4 users active in that frame, a possible scenario could be the next one.

Users	Slot 1	Slot 2	Slot 3	Slot 4
1				
2				
3				
4				

Figure 15. Receiver Example

In the previous figure the slot where a given user transmits is filled with colour. To make this example clarifying, we assume that users are ordered in descendent order as a function of their transmitted power, being user one the one with highest power and user 4 the one with lowest power. Firstly, the receiver will set $N_{iter} = 1$. Let us say the receiver decodes successfully the user's 2 packet in slot 3 and user's 1 packet in slot 4.

So, the decoded packets in $N_{iter} = 1$ are

SLOTS	Slot 1	Slot 2	Slot 3	Slot 4
Decoded Packets	-	-	User 2	User 1

Figure 16. Receiver Example Decoded Packets

Now the receiver cancels all the interferences caused by the already decoded packets, leading to the following updated frame:

Users	Slot 1	Slot 2	Slot 3	Slot 4
1				
2				
3				
4				

Figure 17. Receiver Example Updated Packets

Now the receiver increments the number of iteration ($N_{iter} = 2$) and tries to detect and decode the remaining packets in the frame. The receiver does this process until it reaches the maximum number of iterations defined as $N_{iterMAX}$.

Having explained a simple case. Let us study the mathematical expressions that performs the receiver in order to cancel the interferences.

The received signal **for one slot** follows the expression:

$$r_x(n) = \sum_{i=1}^{K \cdot p} h[i] \cdot b_{s,i}(n) + n(n)$$

When the receiver has finished an iteration (it has tried to detect & decode the packets in all the slots of the frame), it will have a certain number of packets decoded and regenerated in its buffer. **For a certain slot**, let us say that it has decoded successfully N_{recov} packets in a certain iteration. N_{recov} stands for **number of packets recovered in that given slot in a certain iteration**. So, the interference cancelation will be done as follows:

$$r_x(n) = \sum_{i=1}^{K \cdot p} h[i] \cdot b_{s,i}(n) - \sum_{j=1}^{N_{recov}} \hat{h}[j] \cdot \hat{b}_{s,j}(n) + n(n)$$

If we define a new parameter N_{rem} as the number of packets remaining (they have not been decoded yet) in that slot, we can express $K \cdot p$ (the packet in that slot) as:

$$K \cdot p = N_{recov} + N_{rem}$$

Therefore, the received signal can be re-written as:

$$r_x(n) = \sum_{i=1}^{N_{rem}} h[i] \cdot b_{s,i}(n) \; + \underbrace{\sum_{k=1}^{N_{recov}} h[i] \cdot b_{s,k}(n) \; - \sum_{j=1}^{N_{recov}} \hat{h}[j] \cdot \hat{b}_{s,j}(n)}_{\boxed{\text{Error due to channel estimation: } e_j(n)}} + n(n)$$

Then, $r_x(n)$ follows this expression:

$$r_x(n) = \sum_{i=1}^{N_{rem}} h[i] \cdot b_{s,i}(n) \; + \sum_{j=1}^{N_{recov}} e_j(n) \; + n(n)$$

Finally, a state diagram of the receiver is drawn.

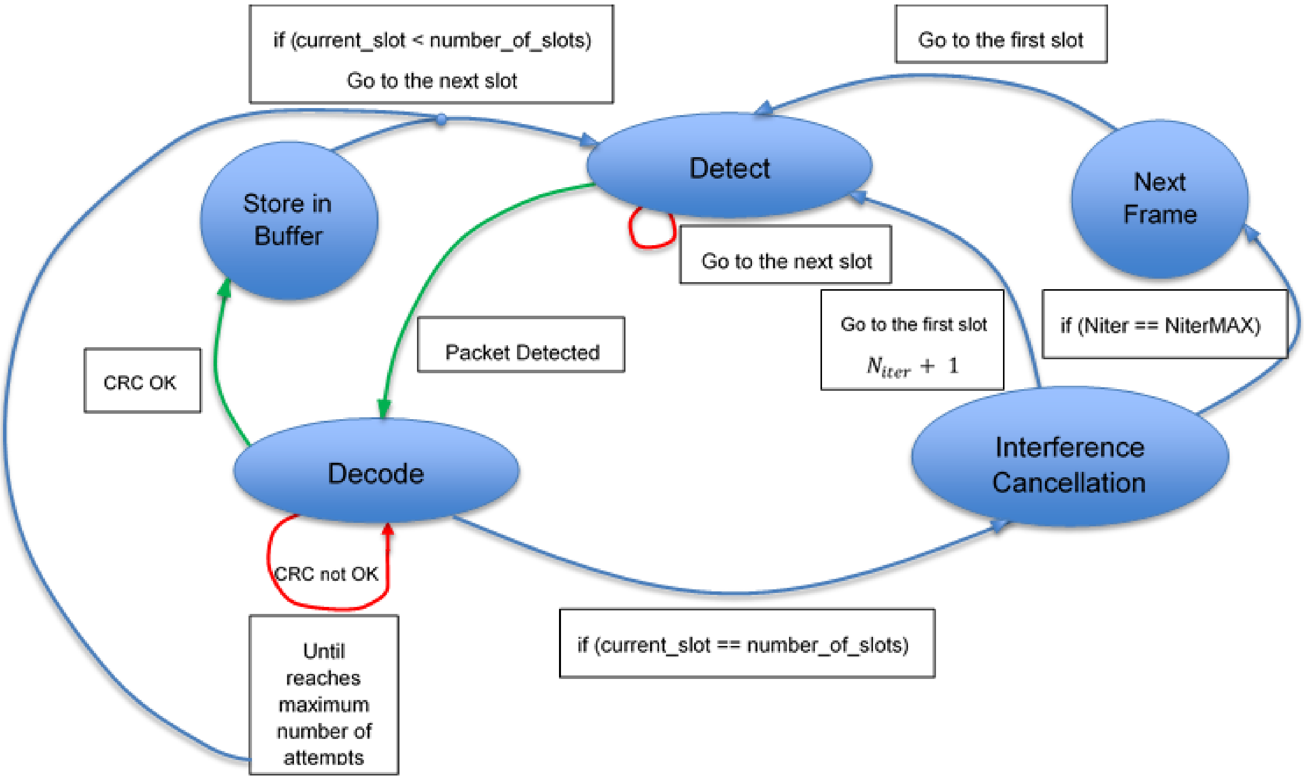

Figure 18. Receiver's State Diagram

4. Results

In this section, the throughput and the packet error rate (PER) obtained simulations are studied. These two parameters are the output of the software designed.

Two main cases have been studied, the ideal channel estimation case and the non-ideal channel estimation case.

Firstly, the parameters under study must be defined:

The Throughput has been defined as

$$T(Kp) = \mathrm{Prob}(Packet\ Successfully\ Decoded) \cdot Kp$$

Where the probability of packet successfully decoded has been computed as:

$$\mathrm{Prob}(Packet\ Successfully\ Decoded) =$$

$$= \frac{\#packets\ decoded(N_{iter} = 1) + \#packets\ decoded(N_{iter} = 2) + \cdots + \ \#packets\_decoded(N_{iter} = N_{iterMAX})}{\#total\ packets\ transmitted}$$

The Packet Error Rate has been defined as:

$$\mathrm{PER} = 1 - \mathrm{Prob}(Packet\ Successfully\ Decoded)$$

The load of the system **G** has been set as Kp , this value represents the number of packets in each slot. So, its units are packets/slot.

Before analysing the results, the background to perform the simulations is specified.

All the simulations have been done considering the following conditions:

$$K = 1000\ users$$

$$p = 3x10^{-3}$$

$$Frame\ Duration\ [slots] = 100\ slots$$

$$N_0 = 1$$

$$Rs = 1x10^3 \frac{bauds}{s}$$

An important parameter to establish, as it has been stated in section 3.3, is the $\frac{E_S}{N_0}$ distribution of all the users in the Frame. An exponential distribution has been chosen to test both the case in which users in the same slot have a similar $\frac{E_S}{N_0}$ and the case in which user's $\frac{E_S}{N_0}$ values differ between them.

The $\frac{E_S}{N_0}$ of each user has been computed as:

$$\frac{E_S}{N_0} = \frac{S_R}{N_0 \cdot R_S}$$

The selected K, p values lead to an **average** load of $K \cdot p = 3\,\frac{packets}{slot}$.

The number of iterations of the SIC algorithm has been increased in each simulation to see the evolution of the algorithm's performance as the number of iterations $(N_{iterMAX})$ increases. The number of iterations set is the following: $N_{iterMAX} = 1, 2, 3, 4, 8.$

The computation of the probability of decoding a packet successfully has been computed using a Monte Carlo analysis. This type of analysis allows us to achieve a mean value of the variable under analysis by performing a statistically representative number of **trials** and computing the mean value of all the realizations. The number of trials to perform the Monte Carlo analysis is set to 50 in all simulations. It has to be taken into account that for each trial, the software analyses 100 slots, which means we analyse 5000 slots with different loads (distributed randomly).

Having explained the simulations' conditions, let us comment the obtained results.

4.1 Ideal Channel Estimation Case

In this case, the regenerated burst that the receiver uses to cancel the interference caused by this burst is exactly the same as the transmitted burst. Thus, no channel estimation error is produced.

The simulated Throughput figure is shown below:

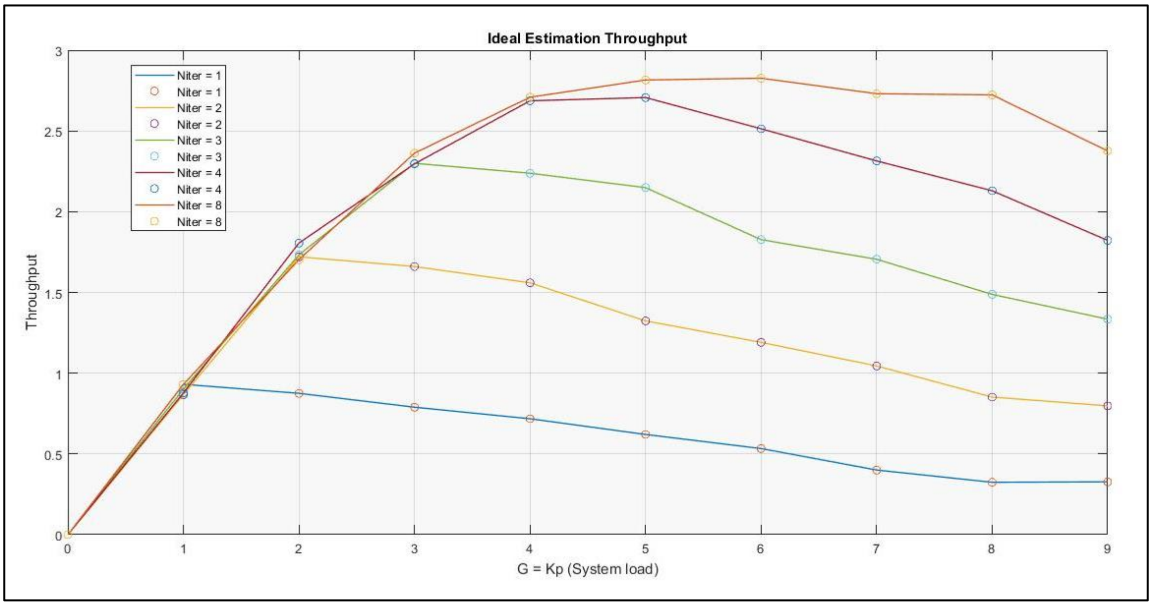

Figure 19. CRDSA Throughput Curves for Ideal Estimation Case

The previous figure shows the CRDSA Throughput with ideal channel estimation. Three main remarks can be done regarding this figure.

1. As expected, the throughput increases with the number of iterations.

2. An important thing to point out is that for each iteration, the Throughput starts to decrease when the channel load is equal to the number of iterations performed by the successive cancellation algorithm. This behaviour is logical because the receiver only analyses a frame $N_{iterMAX}$ times. This means that in the case of $N_{iterMAX} = 1$, the receiver will analyse the frame just 1 time. In this situation, the receiver will decode with high probability the packets in slots with $G = 1$, but since the receiver only analyses the frame 1 time, no cancellation algorithm will be applied. Consequently, in slots with higher load values ($G = 2,3 \ldots$) the receiver will not decode with such a high probability as it does in slots with $G = 1$. In slots with $G > 1$, the decoding success will depend on the signal to interference ratio (SIR) of the users in that slot. To clarify this ratio definition, an example is provided:
   Let us imagine a slot with $G = 3\ packets/slot$. The users are ordered by power in a decreasing way: user 1 (highest power), user 2(middle power), user 3(lowest power). The receiver only will decode the packet if a certain signal to interference ratio in that slot is achieved. Defining this ratio:
$$SIR = \frac{P_{RX}}{\sum_{k=1}^{Nint} P_{RX,k}}$$
   Where P_{RX} is the signal of the user with highest power and $P_{RX,k}$ is the power of the k-th interfering user. For the example provided, the receiver will only decode a packet if:
$$SIR = \frac{P_{RX1}}{P_{RX,2} + P_{RX,3}} > X$$
   This x value has not been computed because was out of the scope of this project, but the previous example and the explanation of point 2 can give us two conclusions about the system:

   - The system can be limited either by noise or either by interferences (in this project the system is forced to be limited by interferences).

   - The system is not near-far resistant. If the SIR is not high enough in a certain slot, the receiver will not be able to decode a packet due to the interferences caused by the users in that slot.

3. Although in point 1 it has been stated that the throughput augments as it does the number of iterations, in slots with large loads it doesn't increase as fast as in slots with low loads. This is because the system saturates. It can be appreciated that the throughput difference between $N_{iter} = 4$ curve and $N_{iter} = 8$ curve is not as large as the difference between $N_{iter} = 4$ curve and $N_{iter} = 3$ curve. This is because the system is saturated, it cannot handle with those loads if the $\frac{E_S}{N_0}$ distribution of the users is exponential as it has been stated in section 3.3. There is a certain number of packets N_{rem} for which their SINR is not high enough to decode them successfully.

Having explained accurately the throughput figure, the PER chart is shown right below:

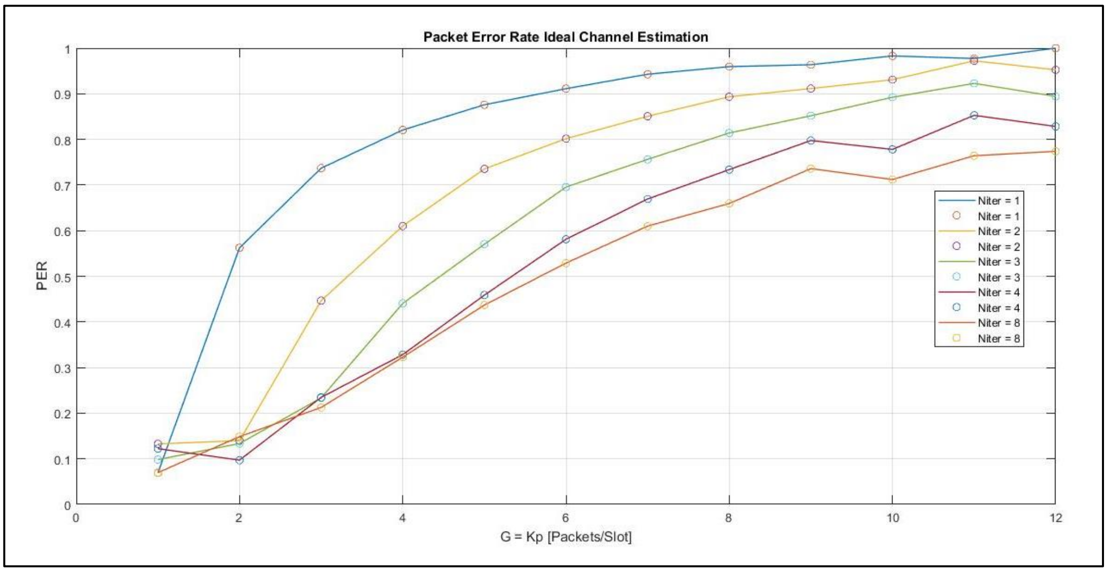

Figure 20. CRDSA PER curves for Ideal Channel Estimation

4.1 Non-Ideal Channel Estimation Case

The non-ideal channel case considers the error due to the channel estimation. Regarding analysis and study of the throughput, no more explanations have to be done because in this case the only thing that changes is that the receiver re-generates the bursts from the channel estimation amplitude and the decoded bits. The throughput curves are shown below for the same number of iterations as well as the PER curve for the same conditions than in ideal channel estimation case. In addition, a comparison between the throughput for non-ideal / ideal channel estimation case are considered.

The Throughput and PER curves for the non-ideal case are the following:

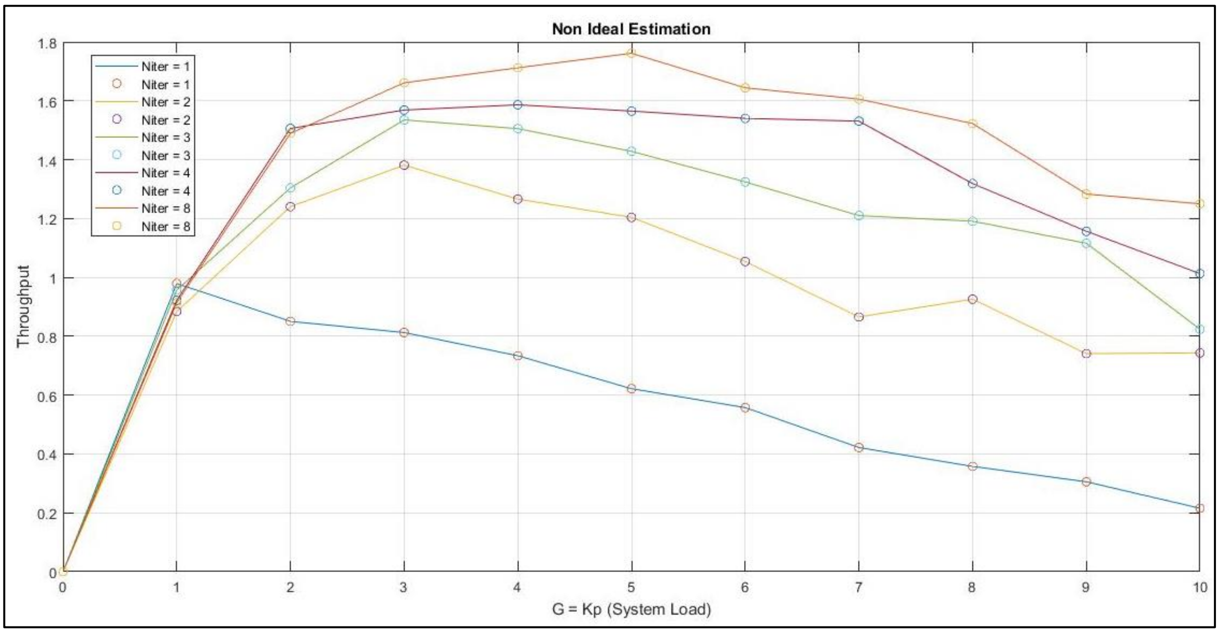

Figure 21. CRDSA Throughput Curves for Non-Ideal Estimation Case

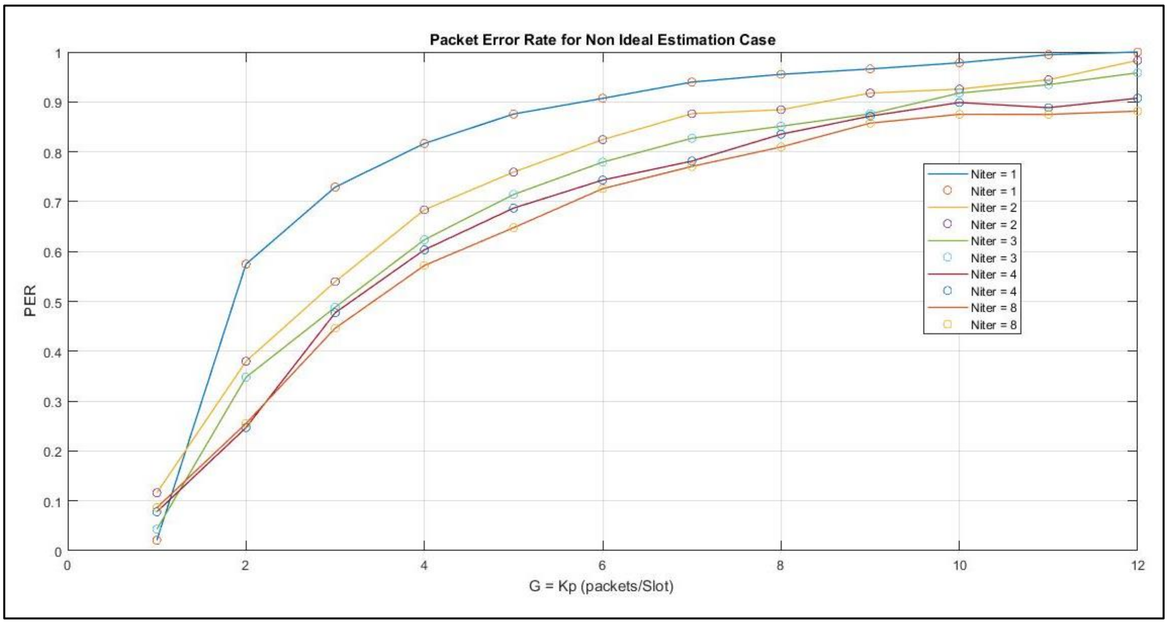

Figure 22. CRDSA PER curves for Non-ideal Estimation Case

To compare the performance of the channel estimator, the following curves compare the throughput for the same number of iterations in non-ideal against ideal channel estimation case:

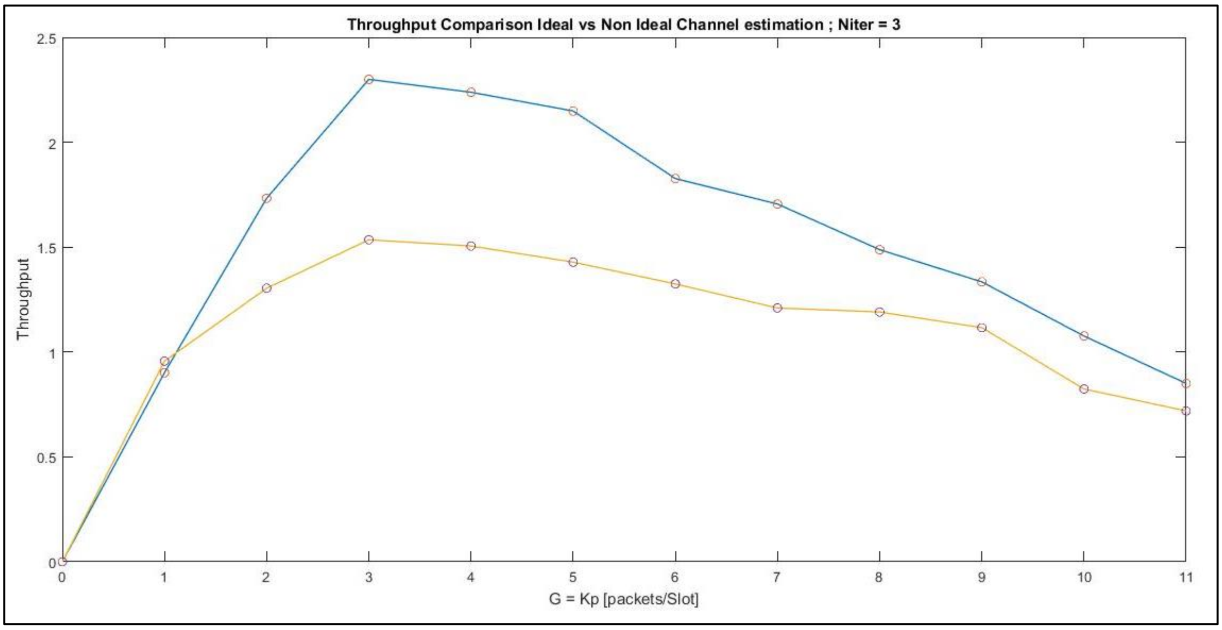

Figure 23. Throughput Comparison Ideal against Non-ideal Niter=3

As can be appreciated in the figure, the channel estimator degrades the performance of the system. A possible solution to this problem could be increasing the length of the preamble sequences to have a better channel estimation.

The case for $N_{iterMAX} = 4$ is shown as well.

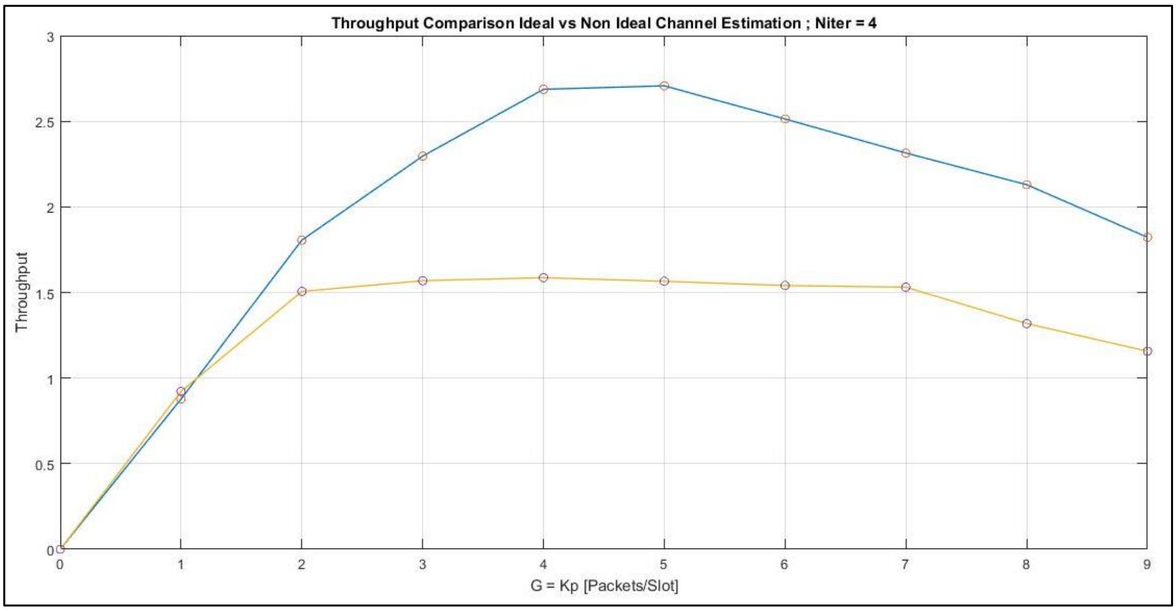

Figure 24. Throughput Comparison Ideal against Non-ideal Niter=4

5. Budget

To approximately calculate the budget of this project, a uniform distribution of the work time has been assumed. This project began the second week of February and will end the 30th of June. From one date to the other there are 18 weeks. The hours/week dedicated had been on average 18 hours/week. The price per hour has been set as 8€/hour. This is the price set by the ETSETB for a junior intern engineer.

Moreover, all simulations have been done with MATLAB®. A MATLAB® license costs 2.000 €. The license provided by the university has been used in this project.

Having explained the budget assumptions, the approximate budget is:

$$Budget = 18\,\frac{hours}{week} \cdot 18\,weeks \cdot \frac{8€}{hour} + 2.000\text{ €} = 4.592\text{ €}$$

6. Conclusions and future development:

As it has been discussed along this project thesis, the performance of the studied system depends on several factors.

The system proposed has shown certain remarkable capabilities, it has shown good throughput curves as the SIC algorithm executed iterations as well as good PER curves. This proposed system has shown a better performance than other RA schemes. However, system's performance can be limited either by noise or by interferences. In this project, it has been assumed that the system was limited only by interferences. This led us to remark a conclusion regarding this topic:

The CRDSA proposed system is not near-far resistant, that is, the receiver reaches a point in which depends on the SIR between the users in the same slot to be able to decode a packet or not.

Regarding future development work, it must be said that all the simulations have been done frame by frame. A sliding window method could be implemented in order to analyse continuously the information sent by the users.

Apart from technical conclusions and future development things, I would like to comment some aspects that have been very important in this project.

In this project no code was provided, so it was begun from zero. This resulted in a very complicated task because all system's features had to be created. Firstly, I had no idea what things I should program first. With a lot of time and research I went coming up with solutions and finally all the parts worked when I put them together. The point I am trying to make is that I strongly recommend to all students who read this thesis to program a system from the very beginning.

Finally, I also would like to comment another aspect that have been relevant in my project. Along the four years of the bachelor degree, I have been learning things from different fields of the Electrical Engineering. Thanks to this project these different concepts have fit in on my mind.

Bibliography:

[1] E. Casini, R. de Gaudenzi, Oscar del Rio Herrero. "Contention Resolution Diversity Slotted ALOHA (CRDSA): An enhanced Random Access Scheme for Satellite Access Packet Networks". IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, VOL 6*, NO. 4, April 2007

[2] Gerard Maral, Michel Bousquet. "Satellite Communications Systems: Systems, Techniques and Technology".*WILEY*, 5th edition, Chapter 4 and Chapter 6

[3] David Tse, Pramod Viswanath. “Fundamentals of Wireless Communications”. Cambridge University Press, 2005. Chapter 10: Multiuser Communication

[4] www.mathworks.com

[5] www.dsprelated.com

[6] TELESP notes, 230257 – Space Telecommunications – ETSETB – UPC, http://www.upc.edu/estudispdf/guia_docent.php?codi=230257&lang=ing