Source: ESA/NASA (2011)

Analysis of an ARQ architecture for Free-Space Optical communications in a LEO-to-Earth scenario

Mafalda Gonçalves Lacerda Teixeira

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Eng. Jorge Pacheco Labrador
Prof. Fernando Duarte Nunes

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Fernando Duarte Nunes
Member of the Committee: Prof. Paulo Sérgio de Brito André

October 2019

Source: ESA/NASA (2011)

Analysis of an ARQ architecture for Free-Space Optical communications in a LEO-to-Earth scenario

Mafalda Gonçalves Lacerda Teixeira

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Eng. Jorge Pacheco Labrador
Prof. Fernando Duarte Nunes

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Fernando Duarte Nunes
Member of the Committee: Prof. Paulo Sérgio de Brito André

October 2019
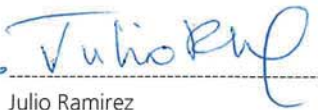
DLR Deutsches Zentrum
für Luft- und Raumfahrt

Non-confidentiality declaration

By the present document, the German Aerospace Center (DLR) confirms that the master thesis report written by Ms. Mafalda Gonçalves Lacerda Teixeira under the title *"Analysis of an ARQ architecture for Free-Space Optical communications in a LEO-to-Earth scenario"* and presented at the Instituto Superior Técnico contains no confidential information concerning DLR projects and knowledge, and therefore can be openly publish.

Oberpfaffenhofen, 29 / 10 / 2019

i.A. _____

Christian Fuchs

i.A. _____

Julio Ramirez

para a Bárbara

Acknowledgments

I would like to thank my supervisor, Eng. Jorge Pacheco Labrador, for first giving me the opportunity to work at DLR and beliving in me throughout the year I was here. Thank you for your patience, all the help and advices, and especially your support.

To Professor Fernando Nunes, I thank your orientation of this thesis on behalf of Instituto Superior Técnico, and your availability.

To all my colleagues, in Lisbon, Beijing and Munich, thank you for giving me the courage to continue pursuing this degree.

To my parents, for all the support. To my sister, for inspiration. For Margarida, for being my family away from home. And to all who have been part of my life in these last five years: thank you.

Resumo

Comunicações ópticas em espaço livre (FSO, em inglês, *free-space optical communications*) podem atingir taxas de transferência superiores a rádio-frequência. O desenvolvimento desta tecnologia oferece novos desafios no mundo das telecomunicações, sendo necessário a conexão enfrentar condições de canal pela atmosfera, diferentes de estudos existentes.

Uma das equipas de investigação de liderança em FSO faz parte do Instituto de Comunicações e Navegação do Deutsches Zentrum für Luft- und Raumfahrt (Centro Aeroespacial Alemão), onde está a ser desenvolvido o programa OSIRIS. Este sistema de comunicações ópticas tem como objectivo alcançar taxas de transmissão de 10 Gbps a partir da Estação Espacial Internacional (numa órbita terrestre baixa) até estações na Terra.

A versão atual deste projecto está a ser desenvolvida com o uso de um pedido automático de repetição (ARQ, em inglês, *automatic repeat request*) para aumentar a confiabilidade da transmissão. O objectivo desta tese foi analisar as possíveis configurações deste protocolo, através de previsões matemáticas da taxa de recepção e a sua validação com simulações.

Diferentes opções foram apresentadas e avaliadas, realçando os pontos fortes de cada sistema e os desafios que expõem. As mensagens de retorno estudadas foram tanto positivas como explicitamente negativas, para um protocolo ARQ híbrido de repetição selectiva. Estes protocolos foram testados com diferentes taxas de transferência e condições de canal (como o índice de cintilação e intervalo de coerência) e optimizados para aumentar a taxa de recepção e diminuir a saturação do canal.

Abstract

Free-space optical communications (FSO) can achieve higher data-rates than radio-frequency. The development of this technology offers new challenges in the world of telecommunications, as channel conditions faced by an optical link in the atmosphere differ vastly from previous studies.

One of the leading research teams in FSO is the Institute of Communications and Navigation at the Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Centre), where the OSIRIS (Optical space infrared downlink system) program is being developed. This optical communications payload aims to achieve a data-rate of 10 Gbps in a link from the International Space Station (in a low-earth orbit) to ground-stations on Earth.

In the current version being developed of this project, an automatic repeat request (ARQ) will be implemented to increase the reliability of the communication system. The aim of this thesis was to analyse the possible configurations of this protocol, by mathematical predictions of throughput and their validation with simulations.

Different options were presented and evaluated, with mention on the highlights of each system, and the challenges that they exhibit. Both positive and negative acknowledgements of Selective-Repeat Hybrid ARQ protocols were studied. They were tested on different data-rates and channel conditions (such as power scintillation index and coherence time) and optimised for a higher throughput and lower channel saturation.

x

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter presents the motivation for this thesis, the environment it is introduced in, and a preliminary survey on the topic. Afterwards, the objectives are introduced, followed by a thesis outline.

1.1 Motivation

The increased demand for reliable communications with high data-rate has trailed the path for Free-Space Optical communications (FSO), a technology that uses light propagation in air and space to transmit information. It is used in situations where it isn't possible to deploy optical fiber, such as air or space borne systems.

Although FSO has been in use since man has communicated with a torch during the night, the path for technology used for FSO was created by Alexander Graham Bell in 1880, by creating the photophone, a device which allowed to transmit voice conversations through the air. Its use became more common only during the 19th century for military applications, as it allowed for undetected communications. With the invention of LASERs (light amplification by stimulated emission of radiation) in the 1960s, revived the investigation in optical communications, even more so in the last 20 years as this method becomes a great alternative to overcrowded radio-frequency (RF) spectrum [1].

In relation to RF transmissions – which use electromagnetic waves with a frequency lower than infrared light–, FSO, also called lasercom, can provide much higher data-rates. The faster transmission of data is not the only characteristic that has made it such an interesting alternative. FSO communications provide larger bandwidth and more spectrum of frequency available. As laser communications imply shorter wavelength, the beam divergence angles are smaller which result in reduced size needed for antennas. The very narrow laser beams used provide an inherent security and robustness to electromagnetic interference. These systems do not require license fees, and have lower installation cost [2].

On the other hand, optical communications have their own disadvantages. The propagation of the laser requires a line of sight for transmission, which means that no data can be received during the passage of a cloud or rainy weather, and multipath reflections can't be used as in frequency-modulation.

Regular atmosphere also interferes with the light beams, lessening the power received due to the scattering of photons. The wavefront distortion is another effect which causes constructive and destructive interference, as figure 1.1 presents.
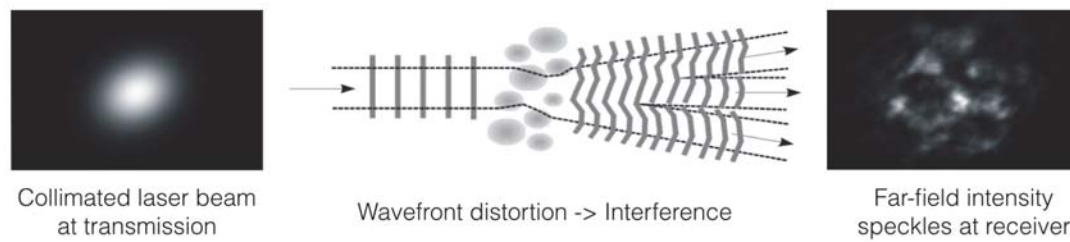
Figure 1.1: Effects of atmospheric interference on a laser.

Lasercom technologies have immense potential of development due to its various applications. It can be used for standard links from low earth orbits (LEO), geostationary orbits (GEO), and deep-space probes to ground stations. Other less common applications are links between LEO and GEO satellites, for earth observation, and networks of airborne or space-borne platforms [3].

Automatic Repeat reQuest (ARQ) protocols send feedback messages to the transmitter to acknowledge for the packets received correctly or ask for missing packets in the receiving end of the communication system. This protocol enhances the reliability of a system by certifying that the packets have been received correctly, or if not, that the transmitter knows so. The implementation of an adequate protocol for the environment it is inserted in will optimise the values of throughput and safely deliver the information to be transmitted.

1.2 Objectives

The analysis of ARQ protocols has thoroughly been studied in the most diverse schemes for communications systems. FSO presents new scenarios and challenges requiring new studies and enabling new research areas. With this thesis, different configurations of ARQ protocols are analysed for specific restrictions in uplink data-rate and reliability. It aims to find the most adequate architecture to achieve an optimal system, i.e., reach a maximum possible throughput with little delay in various channel scenarios.

The results aim to delineate reference values for the transmission while presenting an overview of the improvement achieved with the implementation of the ARQ protocol.

The following thesis is inserted in the research for a transceiver to be implemented in the Bartolomeo platform of the International Space Station (ISS), as demonstrated in figure 1.2. The communication will be provided by a direct link to several ground stations with a data rate up to 10 Gbps over a range of 1500 km [4].

This task is part of the OSIRIS program (Optical space infrared downlink system) which applied the technology of FSO as payloads for small satellites (BiROS – Berlin infra Red Optical System –, and Flying Laptop). The third version, *OSIRISv3* is under development, with aims to achieve data-rates up to 10 Gbps. It will have memory devices which grant the possibility of implementing algorithms to
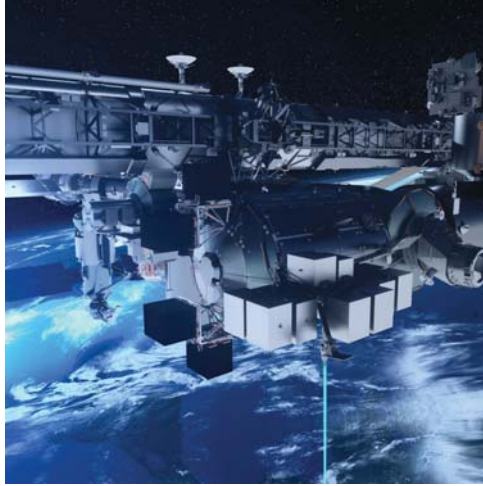
Figure 1.2: Representation of the OSIRIS communication link in the ISS [4].

enhance the reliability of a system, such as ARQ protocols [5] . It is a project included in the Optical Communication Systems group of the Institute of Communications and Navigation which is part of the German Aerospace Centre (DLR – Deutsches Zentrum für Luft- und Raumfahrt).

1.3 Thesis Outline

This thesis begins with some background on the topic, and other theoretical considerations. After that, in chapter 3, a conceptualisation of the analysis and the protocols is introduced. In chapter 4, the simulation model and environment is presented, which is followed by the results in chapter 5. At last, conclusions and future steps compose chapter 6.

Chapter 2

Background

In the following chapter, a theoretical overview is presented on free-space optical communications and an introduction to ARQ protocols, followed by the research conducted on the state-of-the-art of the topic.

2.1 Fundamentals of Communication Theory

The mathematical study of digital communications is called information theory. It includes analytical models of data transmissions, that can be tested using simulations and experiments. Moreover, it can focus on the development of more efficient and reliable systems.

A digital communication system can be defined in a seven layer architecture, by the model ISO-OSI (International Organisation of Standardisation's Open System Interconnection). This model differentiates levels of abstraction, which allows well-defined functions to be set up in each level, from raw signals up to applications such as mail services [6]. Table 2.1 presents this model.

	Layer	Function	Unit
7	Application	holds the applications that act on data (mail services,...)	APDU[1]
6	Presentation	prepares and translates data for the application	PPDU[2]
5	Session	manages and synchronises the conversation	SPDU[3]
4	Transport	breaks the message into small units to be handled by the network	TPDU[4]
3	Network	decides route, divides outgoing data and assembles incoming packets	Packet
2	Data Link	assure sequential and error-free data	Frame
1	Physical	transmission and reception of the unstructured raw data	Symbol

Table 2.1: Open System Interconnection Model.

In the analysis done for this thesis, the ARQ protocol is implemented in the data link layer, as it is a feedback on individual frames.

---

[1]Application Protocol Data Unit
[2]Presentation Protocol Data Unit
[3]Session Protocol Data Unit
[4]Transport Protocol Data Unit

2.2   Free-Space Optical Channel

The channel model of an optical communication system takes into consideration two main influences: the atmospheric turbulence and the pointing error, both which can be modelled by statistical distributions. The analysis of the behaviour of the channel can be done by a Markov error model.

A channel can be characterised by the a set of possible discrete-time inputs and outputs, and the set of conditional probabilities relating to the possible outputs to the inputs [7].

Before any further analysis, some concepts on the parameters used to represent a channel are described in the following subsection.

2.2.1   Channel Conditions

Free-space optical links convey different characteristics from RF channels [8]. These include:

- atmospheric attenuation of laser signals is more severe at low elevation (as the link travels more km, and it's affected by the amount of atmosphere in the path), causing a high variation of received power;

- the link is ofttimes blocked by clouds, resulting in long-term fades;

- the amplitude scintillation patterns of received power are in the order of centimetres (compared to decimetres from RF links), caused by atmospheric Index-of-Refraction turbulence (IRT) – this results in fast fades of optical power;

- the beam in optical communications might be extremely narrow, which can cause an additional source of fading from residual pointing errors of the space terminal;

In order to consider all the attenuation suffered in the path, an approximation can be modelled from example measurements. To combine the effects, a power vector can be used which represents a time series of the received power.

To understand the power vectors' configurations, one has to define a few key terms in communications, which are presented in the following paragraphs.

Data-rate

Data-rate refers to the rate of transmission, i.e., the amount of data transmitted per unit of time. It's usually expressed in bits per second (bit/s). The data-rate measures only the rate of data that is leaving the transmitter, being completely independent of channel errors or losses in the channel.

Power

In order to mimic a real channel for the simulations, the representation of the power received was generated from statistics obtained from experimental campaigns. These values were processed into a normalised power vector which has to be multiplied by a mean power. This way, the probability of error

of the channel can be easily changed, which makes it possible to study the system under different power levels in order to optimise the link budget or account for the ageing of the optical components.

The power is usually expressed in dBm which can be calculated from Watts with the following expression:

$$P_{dBm} = 30 + 10 \log_{10} P_W \tag{2.1}$$

Power Scintillation Index (PSI)

The beam is influenced by atmospheric turbulence (represented by the IRT) along the link path, interfering constructively and destructively with the link, and thus, the power received [8].

The PSI (σ_P) is a common way to describe the fluctuation caused by the turbulence in the power (P), and is determined with equation 2.2.

$$\sigma_P^2 = \frac{\langle P^2 \rangle - \langle P \rangle^2}{\langle P \rangle^2} \tag{2.2}$$

Cloud Coverage

The environment in which this thesis' system is implemented in has to be taken into account, and that includes the meteorological setting it is part of. By considering a single station in southern Europe (more specifically in the Mediterranean region), the link is available between 66% and 84% of times (annual average). The value can be raised towards 100% by distributing multiple ground stations in meteorological uncorrelated locations, which mitigates the cloud coverage [8].

Wavelength

RF waves range from about 20 km to approximately 2 mm. These include radio waves (AM and FM) and microwave bands. Below 2 mm are the millimetre waveband and the infra-red (IR) bands, down until the visible spectrum, which ranges in 0.4 to 0.7 μm. Shorter than these, there are the ultraviolet bands, x-rays, and gamma rays [9].

Figure 2.1 shows the wavelength of the different bands of the electromagnetic spectrum, as well as their penetration in the atmosphere. The wavelength of the laser in use will influence the received power as it will have different capabilities in penetrating of the atmosphere.
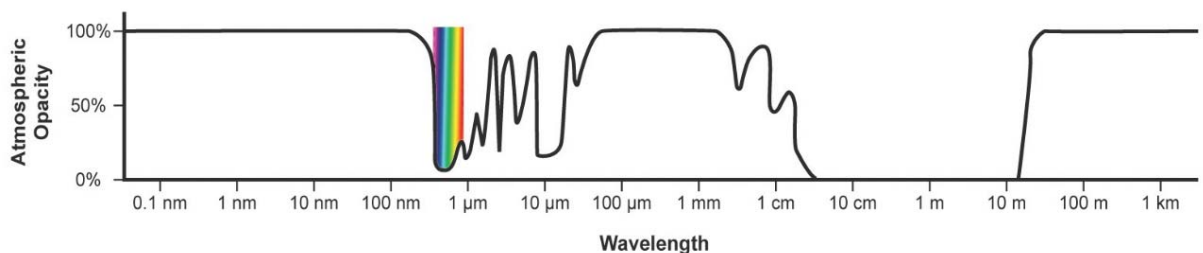
Figure 2.1: Absorption by the Earth's atmosphere of the electromagnetic spectrum. Image credit: NASA.

Channel Coherence Time

Channel coherence time is a measure of temporal coherence. It can be used to quantify the degree of temporal coherence of light, i.e., it is time over which the field correlation function decays [10].

Lasers can have long coherence times compared to an optical cycle, and usually have values in the order of milliseconds. Longer coherence times are usually important for many applications. Throughout this thesis, different coherence times of the channels will be considered and evaluated.

2.2.2   Power Vectors

The power vectors used in the Optical Communications Group at DLR are usually generated from input configurations that are based on experiments. Sometimes, vectors recorded in measurement campaigns can be trimmed and normalised to test an even more realistic setting.

The vectors are generated with input parameters such as the Sampling Frequency [Hertz], the Vector Length [seconds], and the Mean Received Power [Watt] or [dBm]. In order to create the distribution relative to the Atmospheric Turbulence, the desired PSI has to be defined. On the other hand, for the Pointing Error Distribution, the values needed are the Beam Divergence [rad] and the Root-Mean-Square (RMS) Pointing Jitter [rad]. For each distribution, one has to input the Cut-off Frequency [Hertz] and the Low-pass filter (LPF) Slope [dB/decade].

When using the power vector for simulations, it is necessary to use the same vector in order to compare the performance of two different systems. It's not enough to use vectors with the same statistics, as they can contain single rare events that have strong impact on results.

2.3   ARQ Overview

Automatic Repeat Request (ARQ) is a communication technique that aims to improve the reliability of a transmission by ensuring that the message is received by the user.

2.3.1   Basic ARQ Protocols

There are several different protocols which differ in reliability, transmission efficiency and complexity. The three basic ARQ schemes are stop-and-wait, go-back-N, and Selective Repeat [11], and are represented in figure 2.2.

The original and less complex ARQ protocol, stop-and-wait has a basic methodology: the sender transmits one packet at a time and waits for an acknowledgement (ACK) by the receiver. If it doesn't receive that ACK (after a defined timeout) it re-sends the packet. This scheme is limited in efficiency by the round trip delay time [11].

Equation 2.3 gives the formula for the throughput efficiency of the stop-and-wait protocol, with a unitary downlink data-rate (omitted in this section from the throughput formulae) [13].
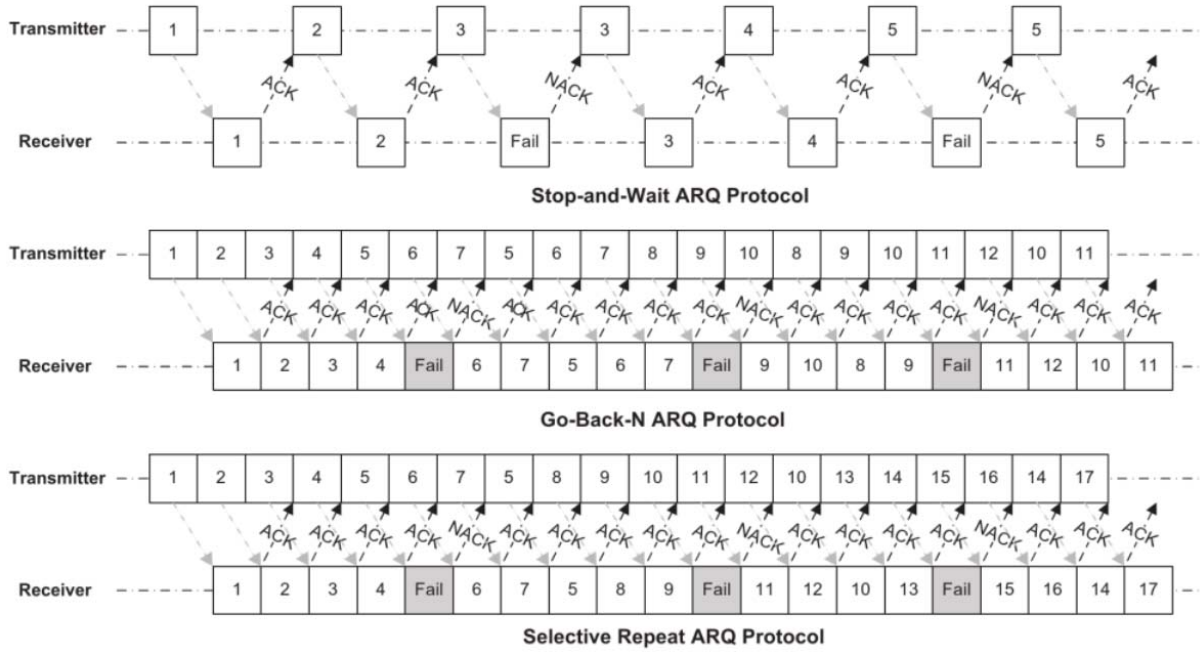
Figure 2.2: Time sequence chart of a the three ARQ protocols presented [12].

$$\eta_{SW} = \frac{P_c \times (k/n)}{1 + \lambda\delta/n} \qquad (2.3)$$

where P_c is the probability that a received word contains no error, k is the number of bits of information in a word of n total bits (including redundancy), λ is the idle time of the transmitter, and δ is the bit data-rate of the transmitter. In systems with very high data-rates, a stop-and-wait protocol will degenerate the performance of the system as $\lambda\delta/n$ becomes very large and a bigger n is impractical to implement [13].

To circumvent the wait time imposed by the stop-and-wait protocol, a window mechanism can be applied, such as the go-back-N. In this scheme, the transmitter has a window of N packets that can be sent without having received an ACK and it advances as ACKs from earlier packets are received. When the window finishes, the sender goes back to the last acknowledged packet and retransmits all of the following ones. This way, the receiver doesn't need a buffer, as it always accepts the packets in order. This protocol is beneficial as it allows for the full use of the data-rate of the transmitter (no waiting time for ACKs). It still loses throughput efficiency as, in the event of an error, the whole window has to be retransmitted [11].

Following the same notation as in equation 2.3, equation 2.4 presents the throughput of a go-back-N protocol [13].

$$\eta_{GBN} = \frac{P_c \times (k/n)}{P_c + (1 - P_c)N} \qquad (2.4)$$

For systems with a high data-rate and long round-trip delay, N can become very large and so, the throughput performance is reduced significantly [13].

It is possible to receive packages out of order by implementing a buffer at the receiver (and the

capability of reordering frames), before delivering to a higher layer. Together with a transmitter that can selectively send frames, one can implement the selective-repeat protocol, where only the lost or erroneous packets need to be retransmitted. This increases significantly the complexity of the system. There can be an implicit retransmission request, where a packet is retransmitted after a timeout (to ensure all packets are eventually received), or an explicit request, where a non-acknowledgement (NAK) is sent by the receiver (which can expedite retransmission) [11].

With respect to the same notation as the previous two equations (2.3, 2.4), equation 2.5 gives the throughput efficiency for an ideal selective-repeat protocol (i.e., with an infinite buffer) [13].

$$\eta_{SR} = P_c \times \frac{k}{n} \tag{2.5}$$

This shows that the throughput in a selective-repeat protocol is independent of the round-trip delay, which is especially beneficial in systems with uplink data-rates much slower than their downlink data-rates.

2.4   Hybrid ARQ Protocols

An improvement on ARQ systems is the use of linear blocks for error control. This method is called Hybrid-ARQ and combines the reliability of the ARQ protocol with the higher throughput performance of implementing Forward Error Correction (FEC) [13].

2.4.1   Forward Error Correction

Forward Error Correction is a technique based on creating codewords with data and redundancy for transmission, which allows the receiver to recover the information under the presence of noise (data corruption) to a certain extent. It can provide some gain for systems with some power limitation, which can be an economic advantage through a compromise on system complexity.

In the systems implemented for simulation in this thesis, all packets sent, in both uplink and downlink, include FEC, therefore, it is important to introduce its concept in order to understand the structure of the messages and calculations of throughput that will show up later on this thesis.

A very efficient group of coding schemes used are Bose-Chaudhuri-Hocquenghem (BCH) codes, which are a subclass from cyclic codes. When a codeword from a cyclic code has a circular shift applied, it creates another codeword. BCH codes' key feature is the simplicity in decoding and the control over the number of correctable errors [14].

The systems in place uses Reed-Solomon (RS) codes, a specific type of BCH, where the locator field is the same as the symbol field. The RS decoding can correct a number of errors of half the size of the redundancy added. As an example, for a redundancy of 32 bytes, the receiver can understand the word with 16 erroneous bytes [14].

There are two types of Hybrid-ARQ: Type-I uses the same code for error detection and correction – useful when a fairly constant level of noise and interference are anticipated in the channel –, and Type-II

works like a basic ARQ when the channel is quiet, but once a retransmission is requested, an extra block of parity-check bits is sent – for channels with a non-stationary bit error rate [15].

2.5 Adaptive Rate Systems

An Adaptive Rate (AR) system is applied at the physical layer to maximise the data rate and satisfy a Quality of Service which is presented as target Frame Error Rate (FER_{target}), i.e., the number of erroneous frames per total number of frames sent [16].

AR systems change the transmission mode according to the channel's state, whether with a change in rate, modulation, error correcting code, or other. The state of the channel can be obtained by processing the number of retransmission requests, by both the transmitter and receiver. Also, this information can be sent from one to the other.

A basic AR applied on a Hybrid-ARQ system is considered, where a two-state channel with different probabilities of channel symbol error are included. For each channel state, there is a corresponding code whose ratio of error correction to error detection is selected to maximise throughput and maintain data reliability [15].
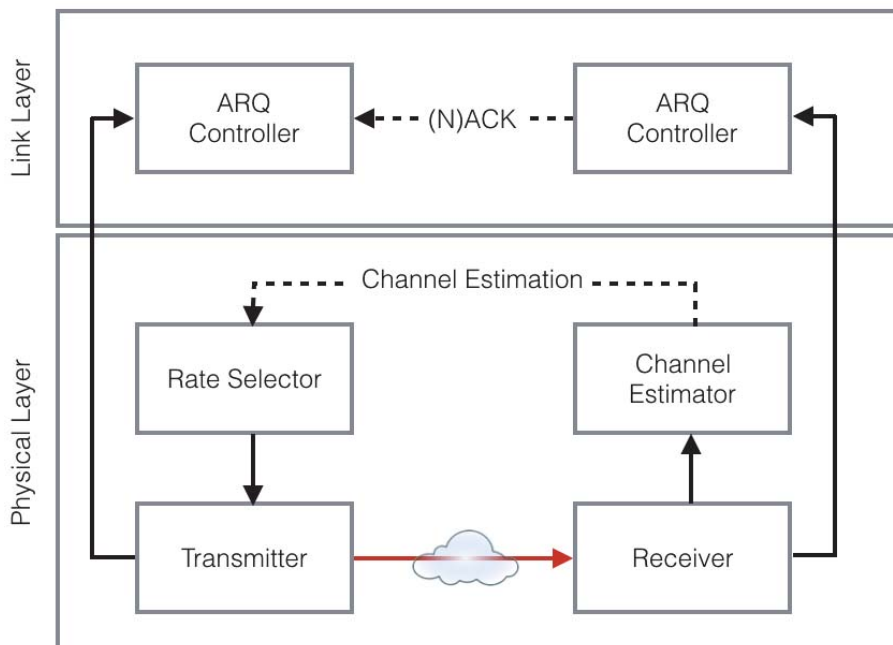
Figure 2.3: Flowchart of a system with adaptive rate control.

AR systems have a significantly better performance than the fixed rate cases for different levels of turbulence, but especially in strong turbulence. It presents a system where the receiver sends a feedback message on the channel state so the transmitter adapts the data-rate (figure 2.3) Mai and Pham [17].

2.6   Uplink Rate Constraints and Reliability

A trade-off between uplink feedback rate and downlink performance has to be considered. To do so, the throughput efficiency is studied with a channel model, defined by an outage probability (the fraction of time that the received power is under a power threshold) and an average non-outage probability. The best throughput efficiency is obtained when the frame-span (transmission time of a frame) is very small compared to the channel coherence time. The required feedback rate is proportional to the propagation time and has an inverse relationship with the frame-span size (T_f). If two terminals are synchronised, and the encoder knows which frame each ACK/NAK corresponds to based only on timing, the sufficient rate is 1 bit/ACK, or $1/T_f$ bit/s [18].

An example is also given for this system in a LEO-to-ground link (where a coherence time of 1 ms is assumed) with a propagation time of 4 ms for a distance of 600 km. To avoid significant loss of throughput efficiency, a feedback rate of at least 10 kbps is necessary [18].

Considering a unreliable feedback channel, with a fairly high ($5 - 10\%$) probability of an unsuccessful transmission, a protocol is proposed where each ACK/NAK carries information about the previous packets, allowing for a lost ACK to be received in following feedbacks (adding a time diversity factor). One packet is only considered lost after either an explicit NAK or a time-out (which includes the round trip and the time for the "extra" possible feedback to arrive) [19].

The numerical analysis allows for the conclusion that an adequate degree of time diversity (i.e., how many past ACK/NAK are sent in one feedback transmission) effectively combats the unreliability of the feedback link. The throughput increases with this degree, which gives rise to a trade-off between efficiency and complexity of the buffer management and time of reassembly.

2.7   Other Considerations

2.7.1   Low-Earth Orbits

Low-Earth Orbits (LEO) can range from 160 km to 1000 km above the Earth. In this kind of orbit, a satellite can take around 90 minutes to circle the Earth [20].

The costs for this kind of orbit are usually lower, as they require smaller launchers. On the down side, a satellite will be moving quite fast relatively to the ground station, and the pointing has to be highly directional in the short period of contact, which can be from 5 to 20 minutes per orbit. In order to have a meaningful data exchange, there must be enough ground stations provided and efficient handing off between the different parties as they connect through the various sites [21].

2.7.2   Orbit Mechanics and Contact Time

Schieler and Robinson [22] analysed the error-free delivery from LEO to a single ground terminal, with consideration for the orbit and its effects on the link losses.

In a LEO-to-ground atmospheric channel, the received power varies in a deterministic manner with the distance of the satellite above the horizon, due to range loss, atmospheric absorption and scattering loss. Adding to these slowly varying losses, scintillation and turbulence-induced imperfect coupling to fiber generate random fluctuations in received power.

For a system with selective-repeat ARQ, where it is possible to achieve error-free transmissions when the received power P_{rx} is above a certain threshold, P_{th}, the throughput efficiency α can be given as a function of the elevation angle θ by

$$\alpha(\theta) = 1 - \mathbb{P}[P_{rx}(\theta) < P_{th}] \tag{2.6}$$

where $\mathbb{P}[P_{rx}(\theta) < P_{th}]$ is the outage probability of the link. With this information, Schieler and Robinson [22] also present the reduction in data rate met by the achievement of reliable, error-free link in

$$R_2(\theta) = \alpha(\theta) \times R_1 \tag{2.7}$$

where R_1 is the transceiver rate and R_2 the end-to-end rate.

Links capable of delivering more than 50 Terabytes per day in a LEO-to-ground scenario have been studied. This is achieved with very high data-rates (>100 Gbps), which will deliver high amounts of information even with a link that is only available for short duration. The total data delivered is proportional to the number of spacecraft passes over the ground stations. For a fixed rate link, the spacecraft has to be observed from the ground terminal above a certain optimum minimum elevation angle. The frequency and duration of these passes will be dependent on the orbit inclination and the terminal's latitude [23].

2.7.3 Pointing Error

FSO communications deal with very narrow beam widths and diverse environmental conditions which are attenuated by control systems. This is a subject in development, as the required accuracy and submicroradian jitter stability are a big challenge in control systems, especially when coupled with real-time and autonomy demands [24].

Pointing errors will result in bursts of lost packets orders of magnitude higher than the channel's coherence time, which can be overcome by ARQ. In the development of optical communication systems, the pointing error can be compensated with a trade-off of beam divergence and pointing accuracy, in order to make this error neglectable.

Chapter 3

Conceptualisation

This chapter presents the structures for the analysis of the ARQ systems and the evaluation criteria that will be later compared to the results.

3.1 Structures for the analysis

Before introducing the protocols in more detail, the structures of the messages are defined for context.

3.1.1 Downlink Message

The downlink message structure is presented in figure 3.1
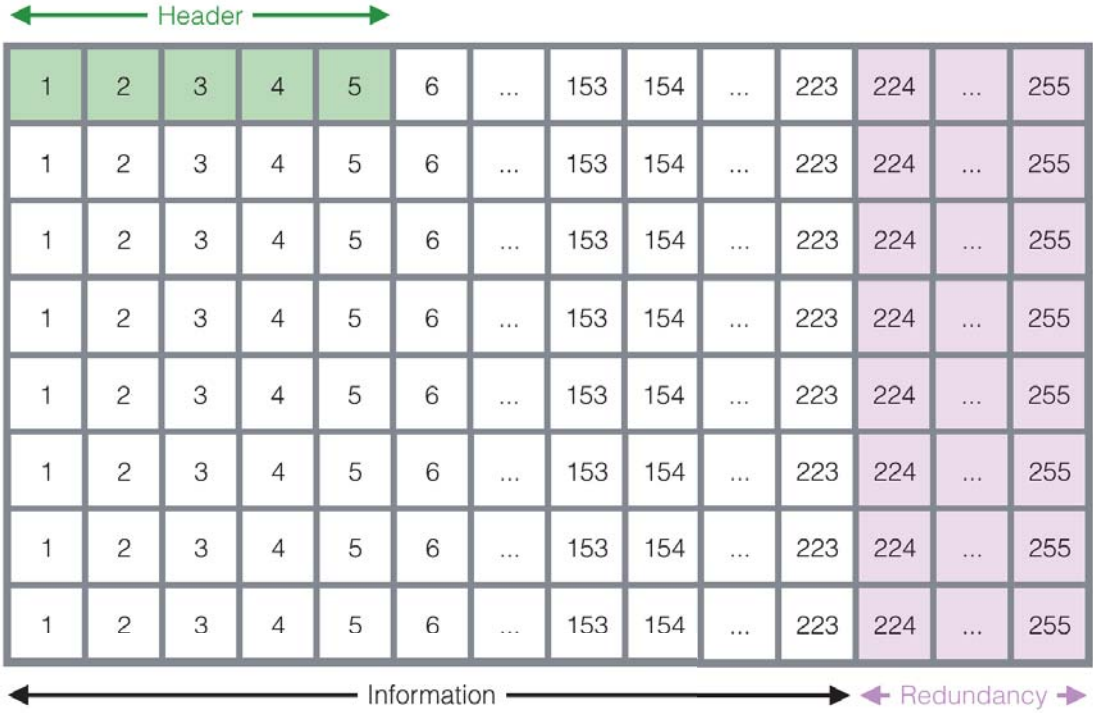
Figure 3.1: Draft of the structure of a downlink message.

One downlink message contains 8 words of 255 bytes each, where 32 are redundancy. The full message is identified by a header of 5 bytes. With a Forward Error Correction scheme, each word can be corrected up to 16 errors, or 32 erasures.

3.1.2 Uplink Message

In our scenario, the uplink message contains 205 bytes of information, encapsulated by 16 bytes of header and 32 bytes of redundancy, following the Consultative Committee for Space Data Systems (CCSDS) standard [25]. That leaves space for 40 "blocks" of 41b to identify the missing frames.
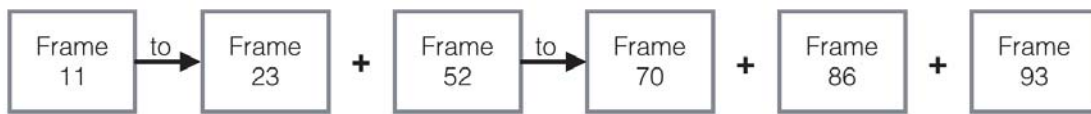
Figure 3.2: Example draft of the block structure of the uplink message (the actual size is of 40 blocks).

In figure 3.2, an example is set for the interpretation of a feedback message. The actual message in numeral representation for the header bytes is presented in the following matrix:

$$\begin{bmatrix} 11 & 23 & 52 & 70 & 86 & 93 \\ 1 & 0 & 1 & 0 & 0 & X \end{bmatrix}$$

where "to" is represented by a bit equal to 1 and "+" is 0. The last symbol of the message is represented as a "don't care" because feedback messages aren't related.

3.2 Channel Model

Using a Markov model for the channel, we can assume $X(t)$ is the fading process: a continuous random process that describes the power delivered to the receiver in the presence of atmospheric fading represented as a log-normal distribution.

When $X(t)$ is less than a power threshold (P_{th}) the channel is in outage. The outage process is defined by

$$X_o(t) = \begin{cases} 1, & \text{if } X(t) \geq P_{th}, \\ 0, & \text{if } X(t) < P_{th}. \end{cases} \tag{3.1}$$

The fraction of time the channel is in outage is denominated the outage probability, p, and any frame received during an outage is erased.

The frame-erasure process of a frame n is defined as $X_e[n]$, and consists of two states, taking the value "1" if the frame is error-free. It is represented by

16

$$X_e(n) = \begin{cases} 1, & \text{if } X_o(t) = 1 \text{ for all } t \in [(n-1)T_f, nT_f], \\ 0, & \text{otherwise}, \end{cases} \tag{3.2}$$

where T_f is the frame-span (transmission time of a frame).

The model $X_o(t)$ is defined as a two-state continuous-time Markov chain (CTMC):

- alternates between two states, outage ("0") and non-outage ("1");

- arrives in state i and stays for a random amount of time;

- the probability distribution of this hold time is an exponential distribution with mean μ_i (average duration of the state);

- the hold time in a state is independent of all past and future holds.

For the defined CTMC with outage probability p and average non-outage duration μ_1, the throughput efficiency α of a selective repeat ARQ is

$$\alpha = (1 - p) \times e^{-T_f/\mu_1} \tag{3.3}$$

Considering τ as the block of time in a constant state of channel condition (coherence time), then $\mu_1 = \tau/p$ and

$$\alpha = (1 - p) \times e^{-p(T_f/\tau)} \tag{3.4}$$

With this model it's possible to observe that a smaller coherence time results in a worse ARQ throughput efficiency [18].

3.3 Types of Protocol

The three different protocols implemented varied on the type of feedback given to the space segment. A basic acknowledgement (ACK) feedback was implemented on the first one, a basic non-acknowledgement (NAK) in the second one, and a Mixed-ACK feedback on the third.

3.3.1 Positive Acknowledgements

In this specific protocol, the ground station sends messages acknowledging the received (non-corrupted) frames.

Acknowledgements (ACK) are an efficient way of ensuring the message was received if the uplink has a high data-rate, so it can inform the satellite in almost real time that the messages are being received.

The method of positive acknowledging the received frames, is advantageous in the sense that the space segment will keep re-sending the not-acknowledged packets until it has confirmation that they

have been properly received. With this system it is ensured that the data arrives to the ground segment regardless of the channel conditions and number of ground segments, given enough time. The control of correctly transmitting information is given to the space terminal.

The studied protocol was based on cumulative-acknowledgements (CACKS), where the received packets in sequence were aggregated to reduce the amount of bits necessary to provide information.

3.3.2 Negative Acknowledgements

The exchange is made by sending messages that report the missing or corrupted frames.

This feedback message is sent by the ground station to the satellite in order to inform the latter about the packages which were not received.

Negative acknowledgements (NAK) operate with the supposition that the data is delivered properly and only re-sends data upon request. In this case, the ground segment has the control of the correctly transmitted information.

The ground station detects the missing packages by checking if there are gaps in the array of saved packets. Then, it sorts them into cumulative non-acknowledgements (CNAKs) by joining packets with consecutive numbers - a few cumulative credentials per packet - as was be represented in 3.1.

3.3.3 Mixed-ACK

The protocol denominated for this scenario as Mixed-ACK relies on a similar structure of the one of the CNAKs, as 38 of the slots are filled in the same way as it, but 2 are reserved for the first and last received packet for synchronisation purposed with different ground-stations.

Between this protocol and the previous, it's predictable that the non-acknowledgements will behave better as the concept is the same but it is less prone to channel saturation. But the decision to use Mixed-ACK is unrelated to this, as this method allows to coordinate with different ground-stations without sharing information between them, which isn't possible with the CNAKs. The Mixed-ACK, the space terminal holds the knowledge of the received data, avoiding cooperation between the ground-stations (that can have its own separate problems). Moreover, having the space terminal control over the data that has been received already, will allow it to delete the data without risk of losing information.

When the satellite reaches a new station it checks the last received packet from the last feedback message of the previous connections and uses a Go-Back-N protocol to send all of the missing packages since then. The new feedback messages received include the first frame delivered to the ground station so the satellite can process if there was any message missing that the new ground station didn't consider for the CNAK.

This is an important part of the connection, which is not considered in the simulations mentioned as it pretends to analyse the ARQ protocol during the ordinary operation time. The influence of this detail is shown in the available number of blocks to signal the NAKs, as it will lead to a slightly bigger channel saturation.

3.4   Evaluation Criteria

This section provides quantitative and qualitative criteria in order to compare the different protocols to be implemented and studied. These terms will be referred to in chapter 5 when analysing the results obtained.

3.4.1   Average Throughput

A high-bound curve for the throughput can be calculated by an expression that takes into consideration the data rate, and the redundancy of the FEC. Equation 3.5 presents this.

η_{max} = throughput high-bound

D_d = data-rate of the downlink

(k, n) = k is the number of information bits for n bits send, according to $\text{RS}(k, n) = \text{RS}(223, 255)$

$$\eta_{max} = D_d \times \frac{k}{n} \tag{3.5}$$

This theoretical prediction can give us a reference value for the throughput to try to optimise our system. This value is only reachable when the probability of error is 0, and no ARQ needs to be implemented.

In order to have a consideration for the losses in the channel, one can induce the formula.

The definition of more variables that need to be included is presented:

p = probability of error in the downlink channel (according to the model of channel used)

q = probability of error in the uplink channel (according to the model of channel used);

T = average number of transmissions per frame.

In order to define the average number of transmissions per frame, the probability of the number of transmissions of a frame is multiplied by that number. For a frame to be transmitted only once, it has to arrive at the first try, which has a probability of $1 - p$. For two times, it has to be lost on the first transmission (probability p), the non-acknowledgement has to be received and the second transmission as well (probability $(1 - p)$), which leads to equation 3.6.

$$T = 1 \times (1 - p) + 2p \times (1 - p) + 3p^2 \times (1 - p) + 4p^3 \times (1 - p) + ... \tag{3.6}$$

By grouping all of the cases in a sum, equation 3.7 is obtained.

$$T = \sum_{i=0}^{\infty} \ (i + 1) \ \ p^i \times (1 - p) \tag{3.7}$$

Considering this sum of infinite terms, its possible to induct it as being a geometric series expansion. With that, equation 3.8 is established.

$$T = \left[\sum_{i=0}^{\infty} (i + 1) \times p^i \right] \times (1 - p) = \frac{1}{(1 - p)^2} \times (1 - p) = \frac{1}{1 - p} \tag{3.8}$$

Equation 3.8 is well know in literature, and the error probability of the uplink is not visible. This happens because when there is an infinite window for the feedback message to be re-sent (which is the method that our system tries to approach), the probability of receiving the uplink message is approximately 1. Equation 3.9 deduces this, where u is the probability of receiving an uplink message.

$$u = (1 - q) + q \times (1 - q) + q^2 \times (1 - q) + q^3 \times (1 - q) + ... \approx \frac{1}{1 - q} \times (1 - q) = 1 \qquad (3.9)$$

Finally, the equation of the throughput, is obtained by dividing the achievable throughput by the average number of transmissions per frame, i.e.,

$$\eta = D_d \times \frac{k}{n} \times \frac{1}{T} = D_d \times \frac{k}{n} \times (1 - p) \qquad (3.10)$$

3.4.2 Throughput as a Function of Time

Since the channel power is time varying, that implies that so is the error probability of a frame. In order to have more significant values of the throughput, an algorithm was generated to obtain the average number of transmissions for a frame generated at a generic time n, which is dependent of future times $(n + 1h, n + 2h, ...)$ with a time-step h which is the round trip delay.

The algorithm presented in equation 3.11 is based on a time dependent function for the average number of frames transmitted. The number of transmissions of a frame generated at time n is "dependent" on the future states of the channel, as its retransmissions always happens steps after the initial transmission.

$$T_n = 1 \times (1 - p_n) + 2 \times p_n \times (1 - p_{n+1}) + 3 \times p_n \times p_{n+1} \times (1 - p_{n+2}) + ... \qquad (3.11)$$

Which can be simplified to:

$$T_n = 1 + p_n + p_n \times p_{n+1} + p_n \times p_{n+1} \times p_{n+2} + ... \qquad (3.12)$$

This time-variant equation will give us a value for the average number of transmissions of a frame generated at time n, which can't be used to calculate the throughput at a certain time. In order to do that, the equation had to be shifted in the time steps.

To create a parallelism to the way the previous equations were obtained, R_n was defined as the number of frames that would be transmitted at time n, with the assumption that the retransmission of frames happens with a certain probability, as explained for equation 3.6.

$$R_n = 1 \times (1 - p_n) + 2p_{n-1} \times (1 - p_n) + 3 \times p_{n-2} \times p_{n-1} \times (1 - p_n) + ... \qquad (3.13)$$

Re-organising this equation for future implementation,

$$R_n = (1 - p_n) \times (1 + 2p_{n-1} [1 + 3 \times p_{n-2} \times (1 + ...)]) \qquad (3.14)$$

The throughput is then calculated with the same formula as in equation 3.10, presented by equation 3.15.

$$\eta_n = D_d \times k/n \times 1/R_n \qquad (3.15)$$

This algorithm can be implemented in *MATLAB*, and repeated for every n in a simulation, which results in values for the throughput in function of time, as well as an average throughput which takes into consideration the time varying channel.

For processing simplification, and to not overload the program, the value for each R_n was not calculated with iterations until the end of the available vector, but until 10.000 samples. The delay variable corresponds to the interval of the time steps and is equal to the time an uplink packet takes to be received. The code shown below is simplified for interpretation purposes.

```
for n = 1 : 1 : length(B)

    R = p(n − delay * max_samples) + 1;

    for i = 1 : 1 : max_samples−1
        R = 1 + R * p(n − delay * (max_samples−i)) * ...
        (max_samples−i+1) / (max_samples−i);
    end

    tp(n) = data_rate * k_n / R;

end
```

3.4.3 Effective Throughput

For simulations, the average throughput is calculated with a direct formula, by multiplying the number of uncorrupted frames by the number of bits per frame, and then dividing it by the simulation time.

The throughput of the frames that can be sent to the higher layer will be equal or lower than the previous, as it stops counting in any "hole" on the array that hasn't been filled. It is a good criteria to evaluate the ARQ efficiency especially with very rough feedback channels.

The frame before the first "hole" is called the *lowbound* and it is the last frame that can be sent to the higher layer, as in order to do so, frames have to be organised, and that can't happen while any is missing.

Figure 3.3 presents an example on how to identify the effective throughput, where the frames in red haven't been received. All the frames previous to the ones in the picture have been correctly received. Then, the lowbound is frame 173 (frame before the first "hole"), and so that's the last frame considered for the effective throughput, even though frames 176 and 177 have already been received.
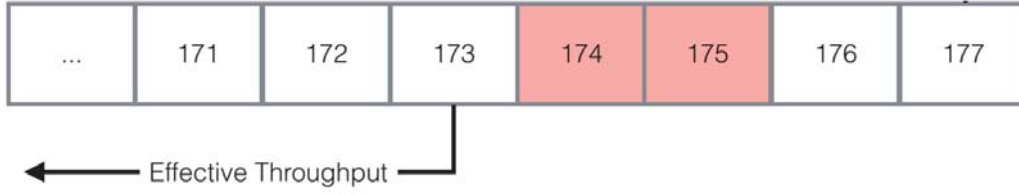
Figure 3.3: Example draft of the effective throughput.

3.4.4 Average Time of Transmission

The average time of transmission is considered the time the satellite spends sending one single frame (throughout the necessary retransmissions). It is inversely related to the throughput, as a smallest average time of transmission will allow to transmit more frames in the same amount of time, which results in a bigger throughput.

To calculate this value, one has to define the time of transmission of a downlink packet,

$$t_{down} = \frac{\text{downlink data rate}}{\text{downlink message length}} \tag{3.16}$$

The formula for this average time is obtained by summing the times spent in each transmission, multiplied by their probability:

$$t_{avg} = t_{down} \times (1-p) + 2 \times t_{down} \times p \times (1-p) + 3 \times t_{down} \times p^2 \times (1-p) + ... \tag{3.17}$$

Grouping all these values in a sum, we get:

$$t_{avg} = \sum_{i=0}^{\infty} \left[(i+1) \times p^i \right] \times t_{down} \times (1-p) \tag{3.18}$$

Using the same method as in equation 3.8, the average time of transmission is obtained:

$$t_{avg} = (1-p) \times t_{down} \tag{3.19}$$

3.4.5 Average Delay Time

Another measure for the evaluation of a protocol is the average delay time. It is an approximation for the time one downlink packet takes to be delivered, obviously being dependent on the number of times it is retransmitted. The delay time considers the time it takes from the first transmission of a frame to its correct delivery.

It is significant in systems with a limited transmission time, although that factor wasn't deeply studied in this thesis and would be an interesting topic for future development.

3.4.6 Channel Saturation

As the feedback channel is much slower than the downlink, it might happen that there is too much information for one uplink frame. When this happens, the channel is considered to be saturated and some acknowledgements/non-acknowledgements might be even more delayed as they have to "wait" for the next message.

This can be overcome with a faster uplink channel or a more efficient way of grouping the feedback messages so more information fits in one packet.

Chapter 4

Implementation

This chapter presents a description of the implementation of the models and algorithms used to represent the system studied. This includes an overall view of the model, the environment for simulations and some specific functions worth mention.

4.1 Overall Simulation Model

The simulation was built by creating modules (the space terminal and the ground station) and channels (downlink and uplink) and simulating their interactions by sending messages. This is represented by figure 4.1. Both modules are ruled by an algorithm that behaves according to the protocol in use. The channels were defined to mimic, as most as possible, the real conditions that a LEO-to-Ground link is subjected to.
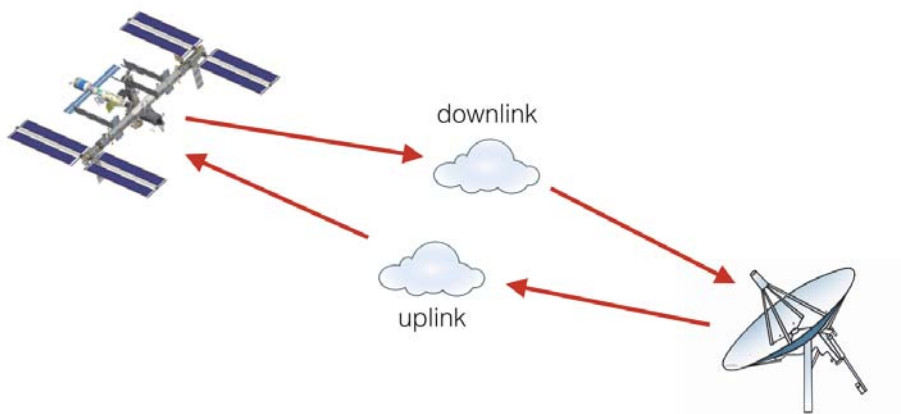
Figure 4.1: Scheme of the communication system.

4.1.1 Ground Station

The ground station module was implemented by creating a processing algorithm for the incoming messages from the satellite and a protocol to request retransmission. It is presented in figure 4.2.
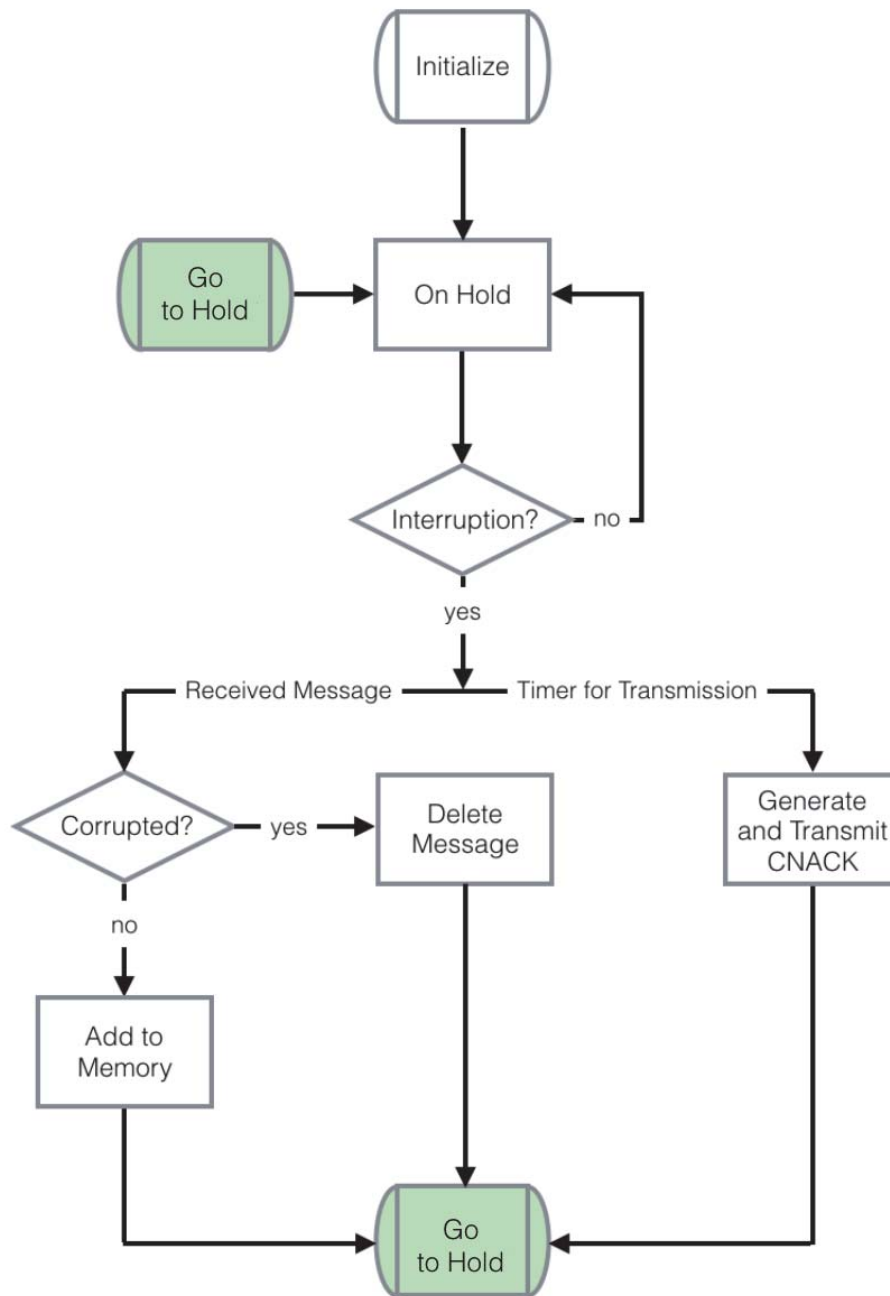
Figure 4.2: Simplified flowchart of the processing algorithm of the ground station.

The received frames are saved in memory (in the form of an array) which enables the receiver to know which information has to be acknowledged/non-acknowledged (depending on which protocol is being used). The highest number of a frame received is called the *lowbound*, as it represents the last frame that can be delivered to a higher layer and forgotten. While there are holes in this array, feedback messages are sent with priority to the lowest numbers, so the information can be given to the higher layer as fast as possible.

The receiver sends a timed self message every few milliseconds (optimised to sync with the transmission time of an uplink message) in order to interrupt itself and check the array and which cells are empty. With that check, the receiver generates either an Acknowledgement or a Non-Acknowledgement

frame and sends it.

To generate a cumulative ACK, the ground-station follows the following process:

1. write the value of the first filled cell of the array in the ACK to be sent;
2. check if the following cell is also filled; if so, the symbol employed is a "to" (if not, the symbol is "+" and step 3 is skipped);
3. run all the cells until it finds an empty cell and write the previous array value in the CACK;
4. look for the next filled cell and restarts the process;
5. stop when either the uplink frame is full or there are no more spaces in the array.

For the Cumulative NAK, the process is similar:

1. the first empty cell is searched for and its value written;
2. if the following frame is also missing, the symbol "to" is written (if not, the symbol is "+" and step 3 is skipped);
3. look for the next filled cell, and write the previous array value in the CNAK;
4. skip until the next empty cell and restarts the process;
5. stop when either the uplink frame is full or there are no more spaces in the array.

To generate the Mixed-ACK, the same process as the Cumulative NAK is employed, except for the the first two slots which are occupied with the first and last received packet in the ground-station.

The change in configuration between the three different protocols is quite simple to implement, as only the generation block has to be altered. If desired, the ground stations could have alternating protocols for various satellites with different configurations, without much cost in complexity.

4.1.2 Satellite

The flowchart in figure 4.3 presents the algorithm of the satellite.

This algorithm is based on two major states: transmission and retransmission. When it is transmitting, it will create new frames and send them at a fixed rate with redundancy. This state can be interrupted with the receiving of an acknowledgement or non-acknowledgement message, which will inform it if any frame should be resent. If an ACK is received, the space terminal will sort the information (received frames acknowledged) into an array with the frame numbers. If there is any gap with missing frames, those values will be selected for retransmission and that state will be active. If the protocol being used sends a NAK, received message is saved as a vector already sorted with the values for retransmission, and that state is activated.

During retransmission state, the space terminal will send all the frames it knows that haven't been received. When it finishes the queue, it goes back to the transmission state. While in retransmission, if it receives a new ACK/NAK, it will update the list according to the new information received with the same procedure as during the transmission state.
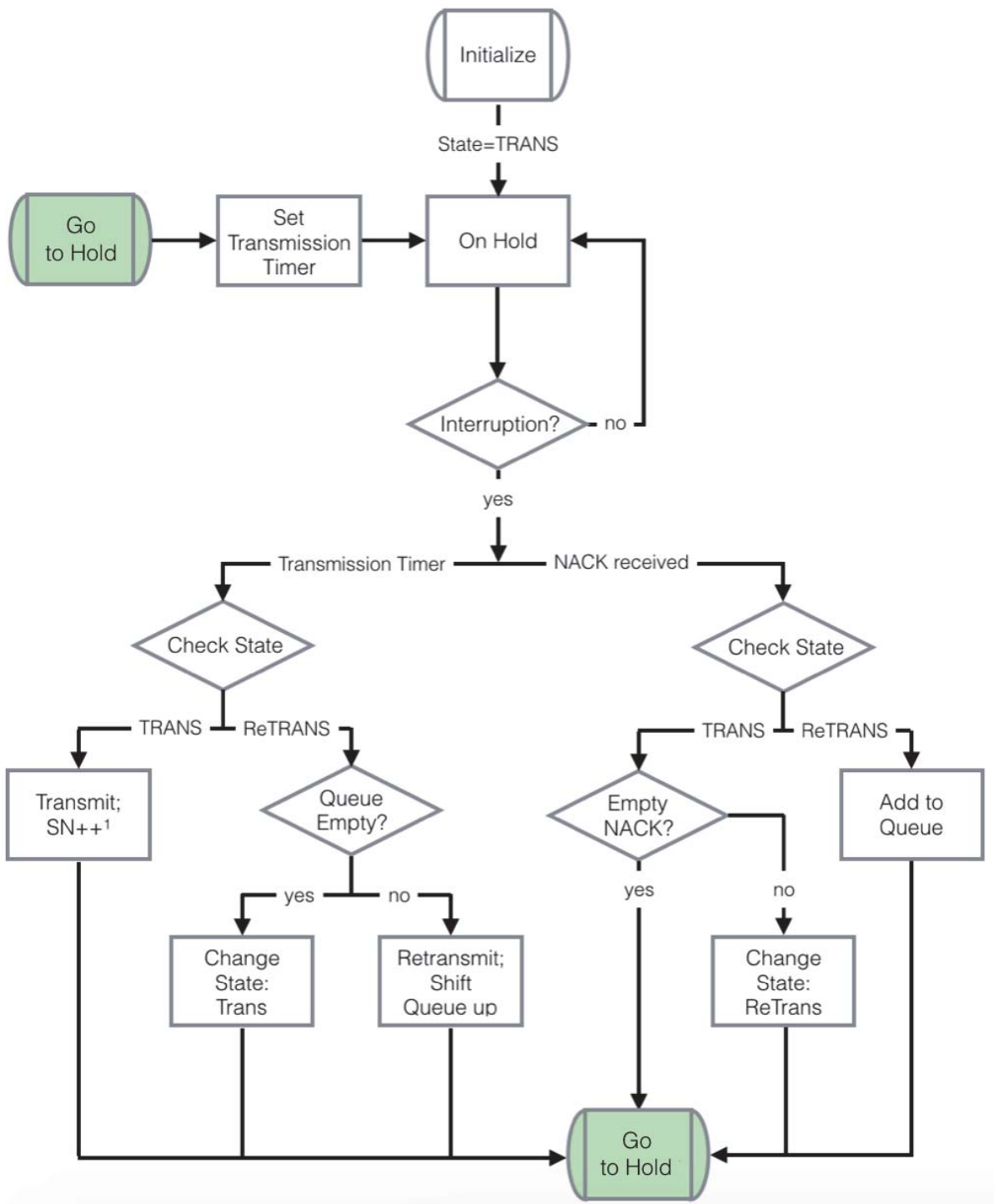
Figure 4.3: Simplified flowchart of the processing algorithm of the space terminal.

For the transmission of a frame, the space terminal receives an interruption, scheduled by a timer – which is set for the time a message takes to be transmitted –, and when this occurs, the following frame to be sent is selected (whether or not it is a new frame or a retransmission is chosen by the state) sent.

It's important that the space terminal has a straightforward algorithm, as over complexity could compromise the speed of transmission. With this method, while fixed on a certain state, there are very few

---

[1]SN = sequence number

delays as most transmissions can be queued up.

4.1.3 Downlink and Uplink Power Vectors

The model of the channel is made by two modules which process the messages in accordance to the statistically generated power vectors to simulate the real atmosphere effect on the link.

Each downlink vector represented is matched with a corresponding uplink vector for a similar atmospheric condition. Although uncorrelated, as they go though different regions in the atmosphere, the links are related by the conditions of the area at the same time (such as a really turbulent or real calm channel). They differ in PSI and mean received power as the links are received through different telescopes with different aperture sizes.

The different situations chosen were based on statistics for measurements at 5° and 15° elevations, which will correspond to the worst and best scenarios, respectively. These elevations were chosen, as only at 5° link connection can be achieved, and above 15° the system is working at a sufficiently high performance that ARQ isn't necessary, and so higher elevations weren't relevant for the scope of this thesis.

Downlink

Four power vectors (A,B, C and D) were used to represent the downlink in two situations chosen – best and worst case scenarios for coherence times of 1 ms and 3 ms. Table 4.1 sorts them.

Vectors	Elevation	PSI	Mean received power	Coherence time
A (best)	15°	0.1	-21.11 dBm	1 ms
B (worst)	5°	0.3	-30.83 dBm	1 ms
C (best)	15°	0.1	-21.11 dBm	3 ms
D (worst)	5°	0.3	-30.83 dBm	3 ms

Table 4.1: Sorting of power vectors by PSI, mean received power and coherence time for the downlink.

For the 1 ms coherence time situation, the best case scenario considered has and elevation of 15° and a PSI of 0.1. For the worst case scenario, a vector was chosen for an elevation of 5° and a PSI of 0.3. The first second of the power vectors are represented in figure 4.4 and the number of fades (i.e., number of times the power lowers 3 dB, 6 dB or 10 dB) in figure 4.5.

For the 3 ms coherence time situation, the vectors chosen have the same PSIs as the previous case, 0.1 and 0.3, for the best and worst case scenarios, respectively. The first second of the power vectors are represented in figure 4.6 and the number of fades in figure 4.7.
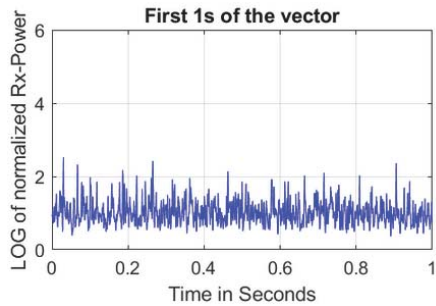
(a) Best Case Scenario - A
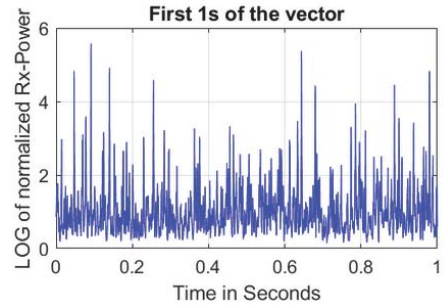
(b) Worst Case Scenario - B

Figure 4.4: Logarithmic plot of the received power during the first 1s of the vectors for downlink (A and B) with coherence time of 1 ms.

(a) Best Case Scenario - A

(b) Worst Case Scenario - B

Figure 4.5: Number of fades in blocks of 10s of 3 dB, 6 dB and 10 dB for the downlink vector (A and B) with coherence time of 1 ms.

(a) Best Case Scenario - C

(b) Worst Case Scenario - D

Figure 4.6: Logarithmic plot of the received power during the first 1s of the vectors for downlink (C and D) with coherence time of 3 ms.

(a) Best Case Scenario - C

(b) Worst Case Scenario - D

Figure 4.7: Number of fades in blocks of 10s of 3 dB, 6 dB and 10 dB for the downlink vector (C and D) with coherence time of 3 ms.

Uplink

In order to represent the uplink, four power vectors (A', B', C', and D'), which correspond to the respective downlink vectors, were chosen for the best and worst case scenarios and coherence times of 1 ms and 3 ms. Table 4.2 sorts them.

Vectors	Elevation	PSI	Mean received power	Coherence time
A' (best)	15°	0.4	-45.06 dBm	1 ms
B' (worst)	5°	0.8	-54.78 dBm	1 ms
C' (best)	15°	0.4	-45.06 dBm	3 ms
D' (worst)	5°	0.8	-54.78 dBm	3 ms

Table 4.2: Sorting of power vectors by PSI, mean received power and coherence time for the uplink.

For the 1 ms coherence time situation, the best case scenario considered has an elevation of 15° and a PSI of 0.4. For the worst case scenario, a vector was chosen for an elevation of 5° and a PSI of 0.8. The first second of the power vectors are represented in figure 4.8 and the number of fades in figure 4.9.

(a) Best Case Scenario - A'

(b) Worst Case Scenario - B'
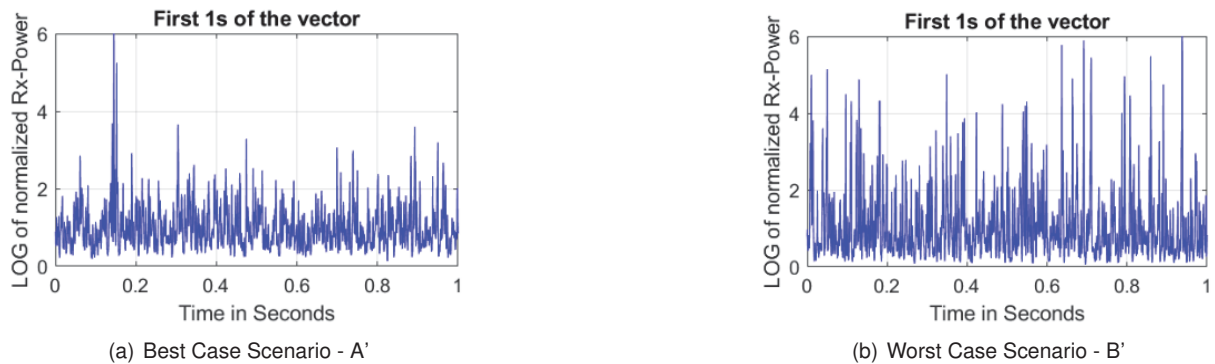
Figure 4.8: Logarithmic plot of the received power during the first 1s of the vectors for uplink (A' and B') with coherence time of 1 ms.

(a) Best Case Scenario - A'

(b) Worst Case Scenario - B'
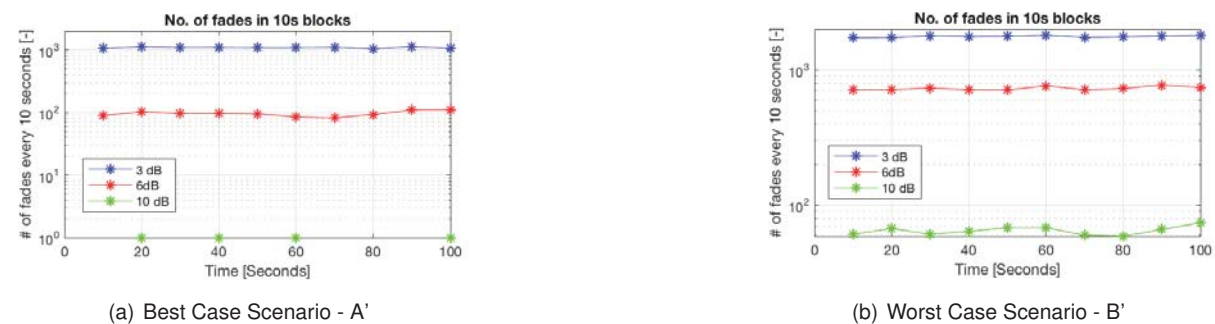
Figure 4.9: Number of fades in blocks of 10s of 3 dB, 6 dB and 10 dB for the uplink vector (A' and B') with coherence time of 1 ms.

For the 3 ms coherence time situation, the vectors chosen have the same PSIs as the previous case, 0.4 and 0.8, for the best and worst case scenarios, respectively. The first second of the power vectors are represented in figure 4.10 and the number of fades in figure 4.11.
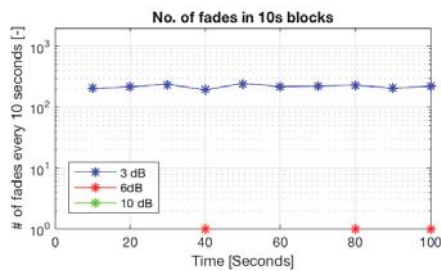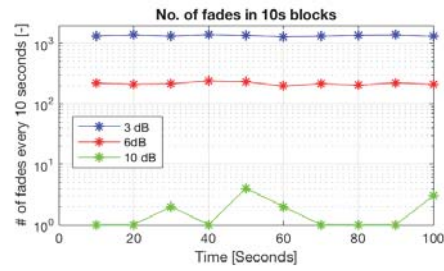

(a) Best Case Scenario - C'

(b) Worst Case Scenario - D'
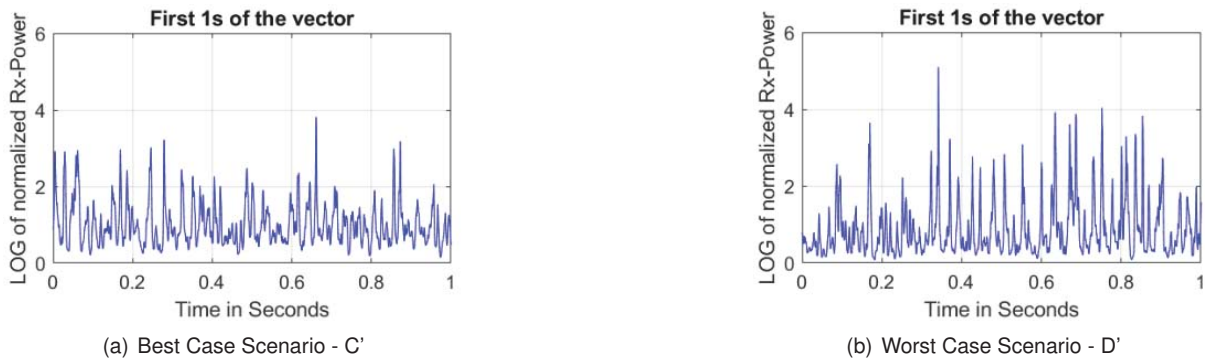
Figure 4.10: Logarithmic plot of the received power during the first 1s of the vectors for uplink (C' and D') with coherence time of 3 ms.
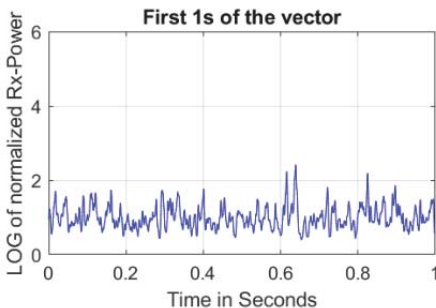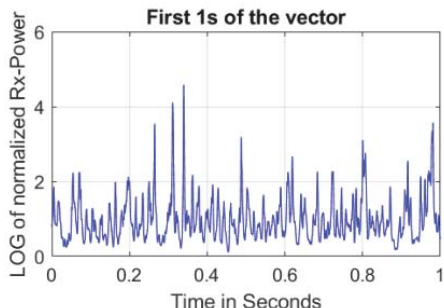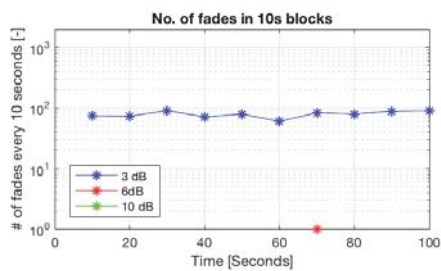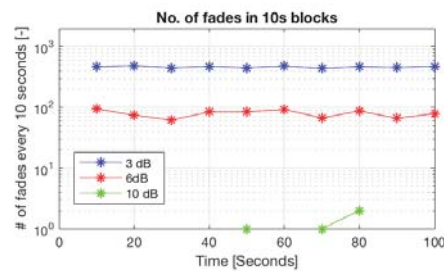

(a) Best Case Scenario - C'

(b) Worst Case Scenario - D'
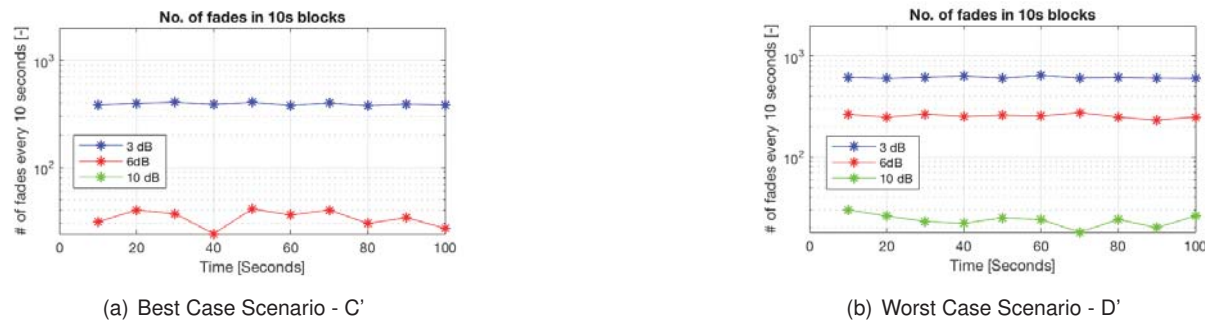
Figure 4.11: Number of fades in blocks of 10s of 3 dB, 6 dB and 10 dB for the uplink vector (C' and D') with coherence time of 3 ms.

The power vector is loaded and multiplied by the mean power used in that simulation. The sampling frequency is also considered in order to sync the vector with the simulation time. There are two possibilities to "corrupt" the message. Firstly, according to a model of the receiver, one can ascertain the Bit-Error-Rate for that sample and calculate the number of erroneous bits in a frame. Then, the simulations could be run with the FEC implemented – which would corrupt non-correctable frames–, or by knowing how many bits it is possible to correct with FEC, and corrupting the frames above that threshold of correctable number of errors. The second possibility, is to define a power threshold above which the messages can be received and corrected and below which they have to be discarded. In this case, the power threshold will have into consideration the gain obtained by the use of FEC (which is around 4 dB), since the specific results of FEC aren't relevant for the simulations being done.

The first option is more precise by taking into consideration the specific number of erroneous bits in a frame, however, due to the random factor of using th bit-error-rate, many more simulations would have to be run in order to have meaningful results. While in the second option only provides an approximation, it is a good choice for the objective of these simulations.

For the formulae conceptualised in the previous section, it was necessary to obtain the probabilities of error in the uplink and downlink of the channel. To formulate a probability of error in a coherence

interval, the number of times the power was under the threshold of correction was counted, and divided by the number of samples in the coherence time. This was done using the following *MATLAB* script:

```matlab
sampling_frequency = 10;          % in kbps
coherence_time = 1;               % in ms
coherence = coherence_time*sampling_frequency;


i=1;
while i<= length(power_downlink_vector)
    count_down=0;
    count_up=0;
    for j=i:1:i+(coherence-1)
        if power_downlink_vector(j) < th_downlink
            count_down = count_down+1;
        end
        if power_uplink_vector(j) < th_uplink
            count_up = count_up+1;
        end
    end

    p(i:i+(coherence-1)) = count_down/coherence;
    q(i:i+(coherence-1)) = count_up/coherence;

    i=i+coherence;
end
```

Figures 4.12 and 4.13 show the power vectors for the best and worst case scenarios for the downlink and the uplink, respectively, with the representation of the probabilities of error of each. These probabilities are obtained by the ratio of time the received power is below the threshold of correction, which is -29.38 dBm in the ground-station receiver (downlink) and -55.40 dBm in the satellite (uplink).

As mentioned previously, the difference between the best and worst case scenarios comes from the different PSI used to generate the vectors, and the mean power by which they are multiplied, that originates from the different elevation degrees that causes lower or higher influence of the atmosphere in the links. As observable in the figures with representation of the received power and threshold of correction, the best case scenario has fewer instances below this line when compared to the worst case scenario. This difference is more evident in the downlink as most of the power vector for the best case scenario is above the threshold, which results in a generally lower probability of error. Besides different PSIs in the downlink and uplink, the normalised vectors are multiplied by an average received power which is -24.87 dBm for the downlink and -48.82 dBm for the uplink.

Figures 4.14 and 4.15 present a similar graphical representation of the previous plots, for vectors

Figure 4.12: Plot of the power vectors for the downlink with coherence time of 1 ms and the probabilities of error, for the best and worst case scenarios in the interval 30–40s.

Figure 4.13: Plot of the power vectors for the uplink with coherence time of 1 ms and the probabilities of error, for the best and worst case scenarios in the interval 30–40s.

with a coherence time of 3 ms.

4.2 Simulation Environment

The simulations were run on *OMNET++* which is a C++ simulation environment, mostly used for network studies. It is based on a component architecture, where modules are programmed in C++, then assembled together by a higher level topology description language: NED.

4.2.1 Interface

Screenshots of the interface of the program are presented in appendix A. In includes the configuration file of a simulation and that same simulation while running. The interactive interface during the simulation allows the programmer to have a look at the actions occurring, which simplifies the debugging process.

4.3 Specific Functions

4.3.1 Packet Generation

The frequency of the packets' generation was also studied in order to optimise the messages sent in the NAK and Mixed-ACK protocols. Another important implementation issue that was considered, is that due to the characteristics of the channel, data should be continuously transmitted in order not to lose synchronisation, i.e., to keep the channel "alive". That way, the generation and transmission of a feedback packet is independent of the number of information that already exist, as the CNAK message is being sent even if not full of relevant information (or even completely empty). Initially, the generation algorithm was called every interval of the uplink send time. Each packet was generated, filled with ordered non-acknowledgements from the oldest to the most recent (and dummy data for the remaining empty blocks) and sent.

Figure 4.16 presents a time sequence scheme where in blue there is a representation of the downlink messages, in green the generation of a new packet, and in red the transmission and reception of an uplink message. The scheme doesn't represent the real data-rates' relationship.

Another possibility studied was to generate two NAKs at a time and only call the generation sequence after the sending time of two feedback messages. That method was saving blocks in the frames as having each NAK generated before sending meant that a feedback wouldn't have still seen the effects of the previous message, i.e., two consecutive messages would request the resend of almost same frames.

By generating two at a time, two consecutive messages will always have different information, correcting the repeated feedback that was happening with the first method. Figure 4.17 outlines this.

35
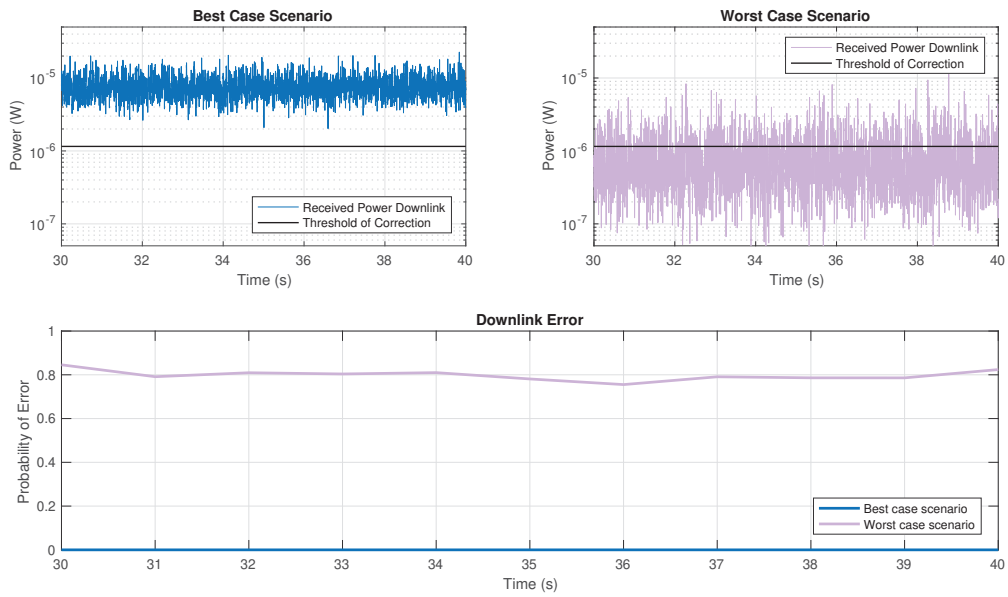
Figure 4.14: Plot of the power vectors for the downlink with coherence time of 3 ms and the probabilities of error, for the best and worst case scenarios in the interval 30 − 40s.
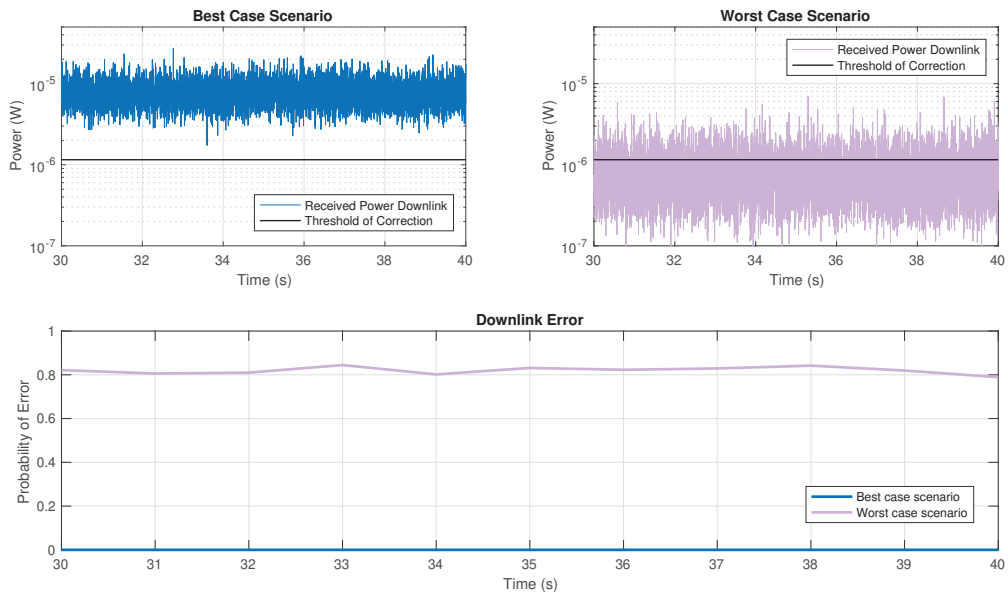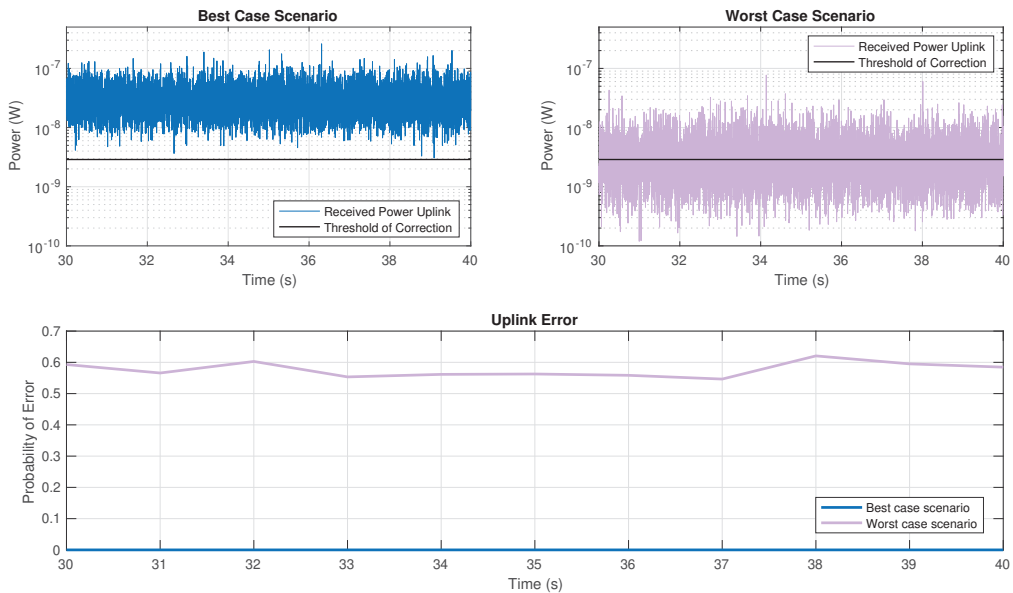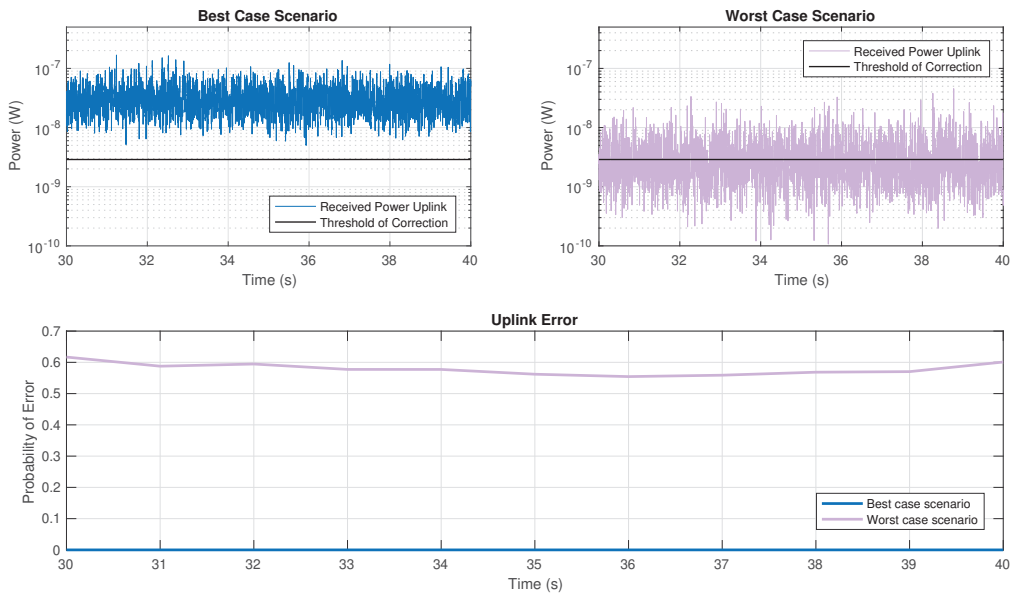

Figure 4.15: Plot of the power vectors for the uplink with coherence time of 3 ms and the probabilities of error, for the best and worst case scenarios in the interval 30 − 40s.
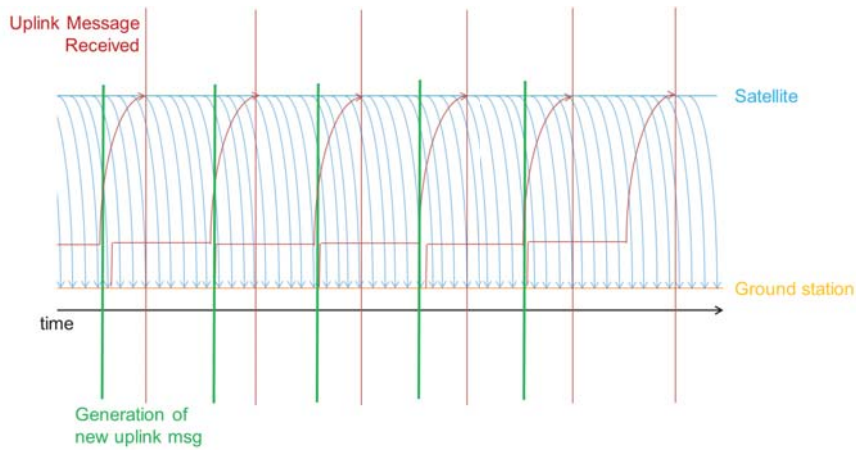
Figure 4.16: Time sequence scheme before optimisation.
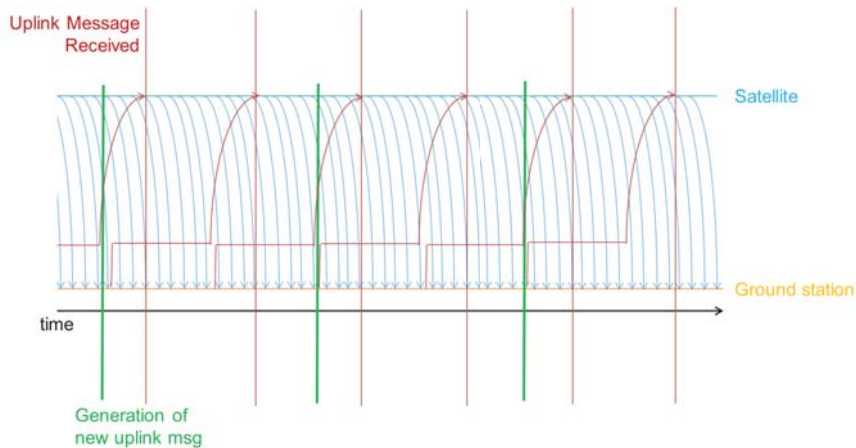
Figure 4.17: Time sequence scheme after optimisation.

4.3.2 Memory Delay

After the ideal system is implemented, the memory access has to be taken into account. In a selective-repeat ARQ, the packets are transmitted non-consecutively, and that delay is orders of magnitude higher than the transmission time of a frame. When packets are sent consecutively, their access is pipelined, so there is no delay to be considered.

For the memory access, a solid-state drive (SSD) was taken into account. The latency assumed for a packet out of order was $35 \pm 20\ \mu s$.

This delay action occurs only when the next frame to be sent is not in the planned order of transmission, i.e., when there is a "jump" in memory. That can happen in three situations:

1. when the state changes from transmission to retransmission;

2. when the state changes from retransmission to transmission;

3. when in retransmission the frames aren't continuous.

Another problem which occurs mostly when there is a channel in the border of correction (i.e., there are short jumps from received to missing frames), is that the uplink channel can get easily saturated. The

cumulative NAKs save a lot of information space for the slow feedback channel, but they are inefficient if there aren't many consecutive messages to aggregate.

Solution Algorithm

One possibility to mitigate the effects of the memory delays is to minimise the jumps in memory during retransmission. This algorithm could be implemented on the satellite or on the generation of the uplink messages. The latter was chosen as it also reduces channel saturation and it's preferred to add complexity in the ground station processing than in the satellite.
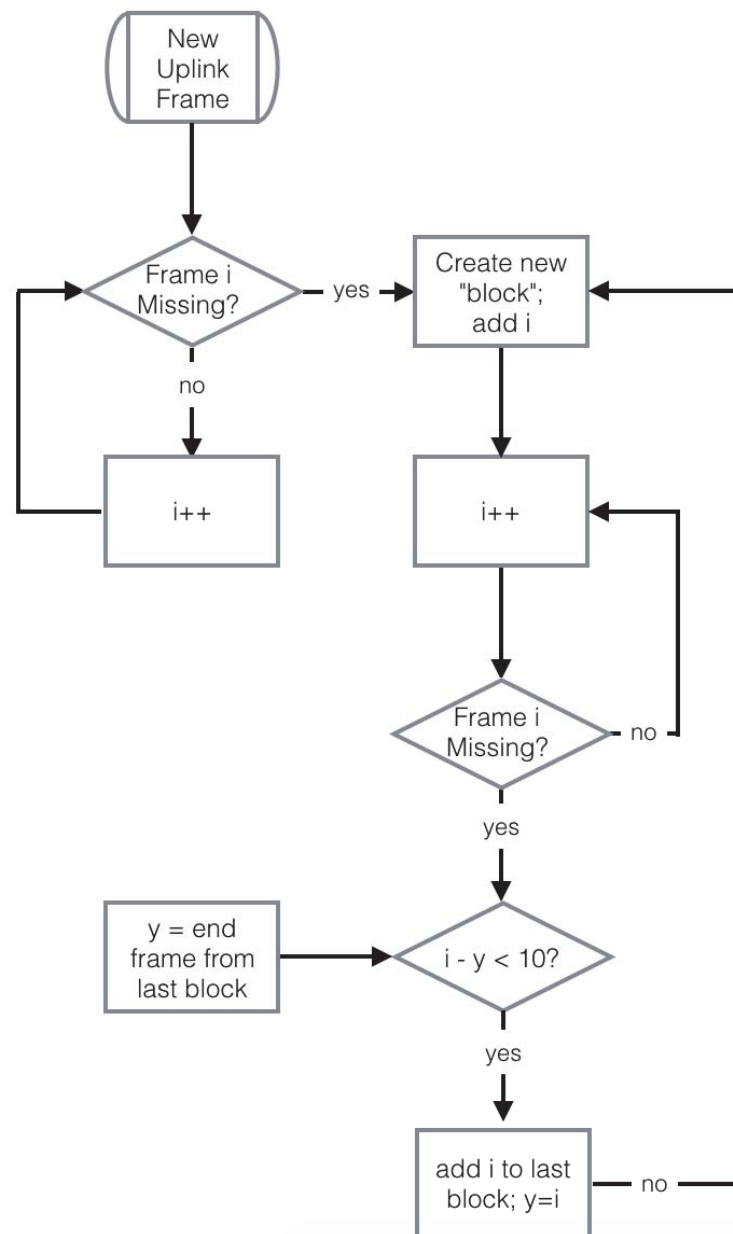
Figure 4.18: Simplified flowchart of the algorithm to mitigate the delay effects of the memory jump, with an example interval of 10.

In this algorithm, when starting a cumulative NAK, the end of the previous NAK is checked. If these

frames have a count distance smaller than a chosen interval, (e.g., interval = 10) then the new cumulative NAK is added to the previous. This is exemplified in figure 4.18.

The algorithm has a variable input which is the value of the interval to be used. Figures 4.19 and 4.20 show how it affects the packets in two different scenarios: a "big" jump and a "small". A "big" jump was called so as it represents when there appear to consecutive "+" symbols in the frame, which means that a value for a missing packet is alone and the retransmission of that data would occur in two memory jumps. A "small" jump simply denominates the cases when two aggregations are close together (separated by a few frames only) and it can be positive to join them in one.

Tables 4.19 give an example of the "big" jump: in the right table the algorithm was applied and it results in **3** less occupied slots (which mean **2** less memory jumps) with the compromise of retransmitting 8 unnecessary frames.

NAKs	
...	...
800	to
850	+
853	+
860	to
1000	X
...	...

NAKs	
...	...
800	to
1000	X
...	...
...	...
...	...
...	...

Figure 4.19: Example of a cut of a feedback packet with a "big" jump, having the algorithm applied in the one on the right.

Tables 4.20 give an example of the "small" jump: in the right table the algorithm was applied and it results in **2** less occupied slots (which mean **1** less memory jumps) with the compromise of retransmitting 3 unnecessary frames.

NAKs	
...	...
800	to
850	+
853	to
1000	X
...	...

NAKs	
...	...
800	to
1000	X
...	...
...	...
...	...

Figure 4.20: Example of a cut of a feedback packet with a "small" jump, having the algorithm applied in the one on the right.

This algorithm reduces the jumps in memory and the delays they cause by sending extra retransmissions. The optimal value for a maximum interval is, then, the number of transmissions that are sent during the same time as the saved memory delay –which is the time of one jump for the "small" case (see tables 4.19) and of two in the "big" case (see tables 4.19).

I_{opt} = optimal interval

D_u = data-rate of the uplink

D_d = data-rate of the downlink

M_d = downlink message size (in bits)

N = number of saved jumps (1 or 2)

T_{mem} = memory delay

$$T_{mem} \times N = I_{opt} \times M_d \div D_d \tag{4.1}$$

$$I_{opt} = \text{int} \left(\frac{T_{mem} \times N \times D_d}{M_d} \right) \tag{4.2}$$

Where int() refers to rounding the number up to the next larger integer. For the values used in the simulations,

$$I_{opt} = \begin{cases} 22 & \text{for } N = 1; \\ 43 & \text{for } N = 2. \end{cases} \tag{4.3}$$

Chapter 5

Results

In this chapter, the values from the simulations are presented and commented, in order to compare the results for different protocols, data-rates, and coherence times, as well as present solutions for a memory delay. A section on the validation of the results is also included.

5.1 Preliminary Results

Primary to presenting the results, table 5.1 sorts the power vectors used for representing the best case scenario, i.e., for an elevation of 15°, where the subscript, X', identifies the uplink vector correspondent to the downlink vector, X. For reference, the threshold considered for the downlink was -29.38 dBm, and for the uplink -55.40 dBm.

Vectors	PSI	Mean received power	Coherence time
A	0.1	-21.11 dBm	1 ms
C	0.1	-21.11 dBm	3 ms
A'	0.3	-45.06 dBm	1 ms
C'	0.3	-45.06 dBm	3 ms

Table 5.1: Sorting of power vectors by PSI, mean received power and coherence time for the best case scenario.

Table 5.2 sorts the vectors for the worst case scenario, with an elevation of 5°. For reference, the threshold considered for the downlink was -29.38 dBm, and for the uplink -55.40 dBm.

Vectors	PSI	Mean received power	Coherence time
B	0.4	-30.83 dBm	1 ms
D	0.4	-30.83 dBm	3 ms
B'	0.8	-54.78 dBm	1 ms
D '	0.8	-54.78 dBm	3 ms

Table 5.2: Sorting of power vectors by PSI, mean received power and coherence time for the worst case scenario.

Going back to equation 3.5, for a simulated downlink data-rate of 10 Gbps and $k/n = 1779/2040$, the maximum throughput achievable for the system is 8.72059 Gbps and will be used as reference for the rest of this chapter.

5.1.1 Results for the Different Protocols

As an initial test, the three different protocols were tested. Tables 5.3 and 5.4 show the values of throughput obtained for the following configuration:

- power vectors with 1 ms of coherence time;
- simulation time of 100 s;
- downlink data-rate of 10 Gbps;
- uplink data-rate of 1.5 Mbps.

In table 5.3, the best case scenario is used for the simulations – an elevation of 15° is considered, represented by the vectors A and A' (downlink and uplink).

Protocol	No ARQ	ACK	NAK	Mixed
Average Throughput [Gbps]	8.72058	8.72058	8.72058	8.72058
Effective Throughput [Gbps]	8.72058	8.72058	8.72058	8.72058
Average Trans. Time [μs]	1.63200	1.63200	1.63200	1.63200

Table 5.3: Comparison of throughput and average transmission time for different protocols for the best case scenario.

In this table, the values shown are present for the best case scenario of atmospheric turbulence. As even without an ARQ protocol in place the throughput reached was (in practice) maximum, one can conclude that for the best scenario no packets are lost, i.e., even if the channel produces errors in the transmission, the FEC system in place can correct them all. In order to study the difference between the ARQ protocols present, the simulations have to be run with more degraded channels.

In the following table, the same comparison is shown with the same configurations except with the power vectors correspondent to the worst case scenario, i.e. an elevation of 5° – using vectors B and B' (downlink and uplink).

Protocol	No ARQ	ACK	NAK	Mixed
Average Throughput [Gbps]	1.68018	1.67995	1.67958	1.67966
Effective Throughput [Gbps]	0	0.26533	1.31694	1.31703
Average Trans. Time [μs]	8.47053	53.4889	10.5624	10.5393

Table 5.4: Comparison of throughput and average transmission time for different protocols for the worst case scenario.

The average throughput of the system without ARQ is the highest achieved as in this case the transmitter never wastes time with retransmissions. Considering the effective throughput, the benefits of the ARQ system are evident, whichever the protocol chosen.

The ACK protocol behaves worse than the rest as it requires the acknowledgement of the packets in order to send new ones. If it doesn't receive it, it will re-transmit packets that were already received correctly, damaging the effective throughput. Although this protocol is efficient in many scenarios, it is limited by the uplink data-rate, which in this case, is significantly lower than the downlink.

The protocols with best performance for the system are with negative acknowledgements, the Mixed being chosen for the synchronisation factor explained before.

5.1.2 Results for Different Uplink Data-rates

In the project, the uplink data-rate is yet to be defined and so the simulations were tested using the different possibilities for the NAK protocol. The following configurations were used for the tables 5.5 and 5.6.

- power vectors with 1 ms of coherence time;
- simulation time of 100 s;
- downlink data-rate of 10 Gbps;
- uplink data-rate of 15 kbps, 150 kbps, 1.5 Mbps.

For the best case scenario, selecting the vectors A and A', the following values were obtained.

Uplink data-rate	15 kbps	150 kbps	1.5 Mbps
Average Throughput [Gbps]	8.72058	8.72058	8.72058
Effective Throughput [Gbps]	8.72058	8.72058	8.72058
Average Trans. Time [μs]	1.63200	1.63200	1.63200

Table 5.5: Throughput and average transmission time for different data-rates for the best case scenario.

No comments can be made for the effect of different uplink data-rates for the best case scenario, as all the values obtained were the same. This happens as for the best case scenario, no downlink packets are lost and the ARQ protocol isn't put in use, so the uplink data-rate has no influence.

On the following table, the same comparison is shown with the same configurations except with the power vectors correspondent to the worst case scenario, i.e., B and B'.

Uplink data-rate	15 kbps	150 kbps	1.5 Mbps
Average Throughput [Gbps]	1.68017	1.68011	1.67966
Effective Throughput [Gbps]	0.15235	1.15658	1.31703
Average Trans. Time [μs]	16.9932	11.9247	10.5393

Table 5.6: Throughput and average transmission time for different data-rates for the worst case scenario.

The effective throughput is higher with a faster uplink data-rate, as could be expected. If faster feedback links were available, it would be possible to achieve effective throughput closer to the average one, as the ideal ARQ protocol allows the satellite to have real time information about the received packets which doesn't happen with a rate limited uplink.

For the uplink data-rate of 15 Mbps, the throughput achieved is 15% of the maximum throughput with a perfect channel. This sets a lower bound in the performance of the system.

5.1.3   Results for Different Coherence Times

In order to study the effects of the correlation time of the channel, the values for throughput obtained for power vectors with 1 ms coherence time were compared to the values obtained for vectors with 3 ms coherence time. The vectors used were previously sorted in tables 5.1 and 5.2.

These values were summarised in three tables, each representative of simulations with different uplink data-rates. The configurations were the following:

- power vectors with 1 ms (A, B) and 3 ms (C, D) of coherence time;
- simulation time of 100 s;
- downlink data-rate of 10 Gbps;
- uplink data-rate of 15 kbps (table 5.7), 150 kbps (table 5.7), 1.5 Mbps (table 5.7).

Scenario	A	C	B	D
Average Throughput [Gbps]	8.72058	8.72058	1.68017	1.65589
Effective Throughput [Gbps]	8.72058	8.72058	0.152355	0.70671

Table 5.7: Throughput for power vectors with different coherence times with an uplink data-rate of 15 kbps.

Scenario	A	C	B	D
Average Throughput [Gbps]	8.72058	8.72058	1.68011	1.65583
Effective Throughput [Gbps]	8.72058	8.72058	1.15658	1.21191

Table 5.8: Throughput for power vectors with different coherence times with an uplink data-rate of 150 kbps.

Scenario	A	C	B	D
Average Throughput [Gbps]	8.72058	8.72058	1.67966	1.65553
Effective Throughput [Gbps]	8.72058	8.72058	1.31703	1.08587

Table 5.9: Throughput for power vectors with different coherence times with an uplink data-rate of 1.5 Mbps.

As predicted in section 3.2, a longer coherence time will theoretically result in a better throughput efficiency. In the worst case scenario, for the uplink data-rates of 15 kbps and 150 kbps, the effective throughput achieved values 4 times and 5% higher, respectively. This improvement didn't happen to the values in the final case. There are some possible justifications for this, as vectors B and D were generated to have the same PSI and were used with the same mean power, they don't behave identically over time. At a rate of 1.5 Mbps, the transmission time of an uplink packet is in the same order of

magnitude as the coherence time, and that can also have a peculiar effect in the results. A deeper approach on this topic is a prime concern for future work.

5.2   Validation of the Results

For the validation of the results, equations 3.14 and 3.15 were implemented in *MATLAB* with the algorithm presented in that section, and plotted alongside the vectors of throughput obtained from simulations. Figure 5.1 presents their plots over time, for the theoretical throughput, the throughput simulated in OMNET++ and the effective throughput, also from simulations.
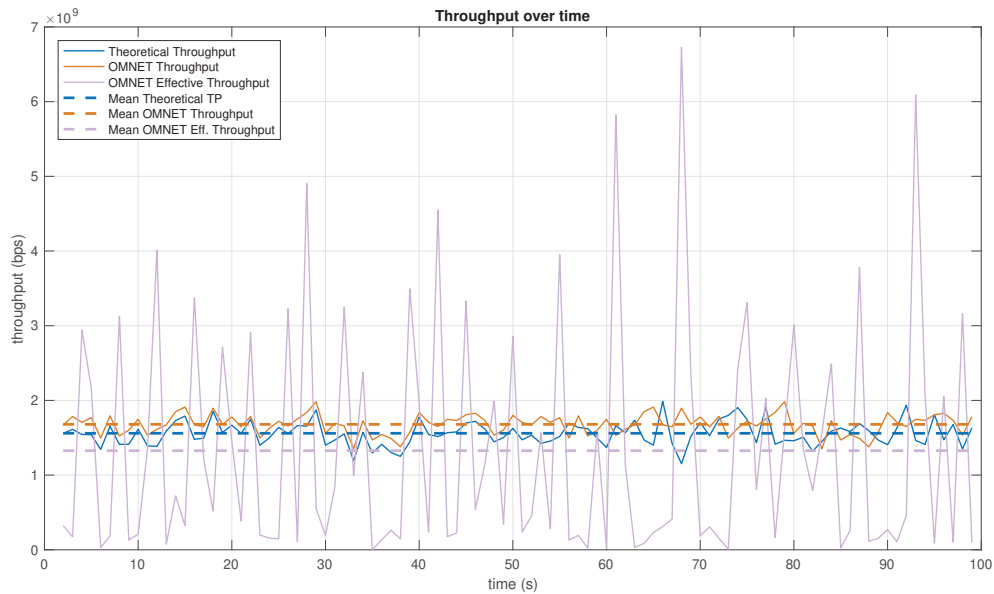
Figure 5.1: Plot of throughput over time for theoretical prediction and OMNET++ simulation (average and effective); vectors B and B' used with an uplink vector of 1.5 Mbps.

The effective throughput has a higher variance than the other two, as it is dependent on the check of the array. In other words, the value for the lowbound is only updated when the memory of the satellite is checked and so there are time periods where its evolution is very small and time period where it's very high.

Figure 5.2 removes the effective throughput for clarity in the comparison of the average throughput (in theory and simulation) and figure 5.3 which is zoomed in for better visualisation.

The theoretical prediction aimed to estimate the value of the throughput by calculating the probabilities of error in the channel and estimating the behaviour of the system in each time step. On the other hand, the simulation ran the transmission and reception of packets over time, corrupting the ones that were being sent while the channel was in outage. For the OMNET++ simulation, the values obtained are slightly higher than the theoretical throughput.

It's clear that the theoretical prediction can accompany the real results obtained, especially in the beginning. This is caused by the fact that as the time passes, more time factors have to be considered
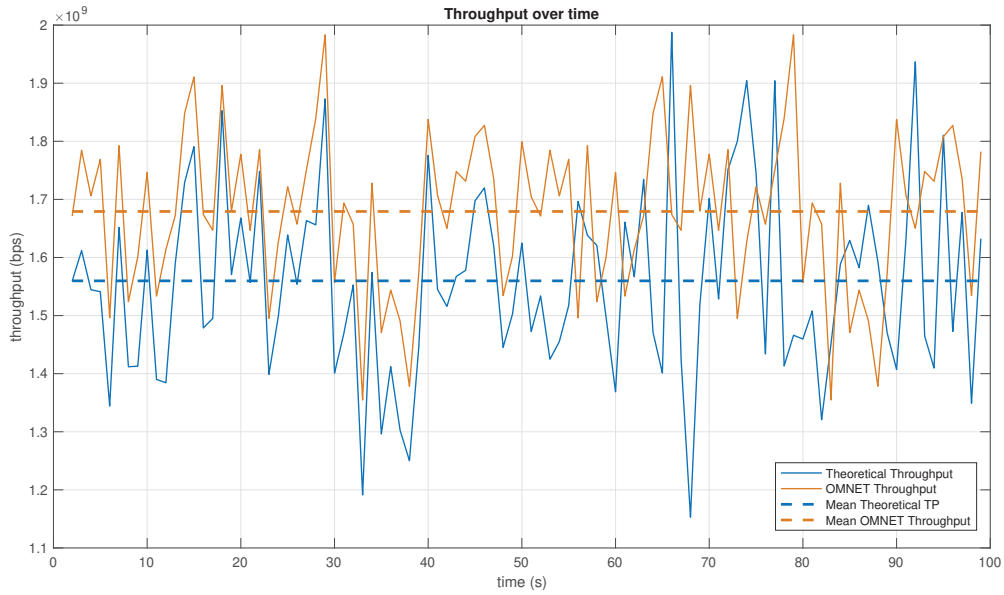
Figure 5.2: Plot of throughput over time for theoretical prediction and OMNET++ simulation; vectors B and B' used with an uplink vector of 1.5 Mbps.
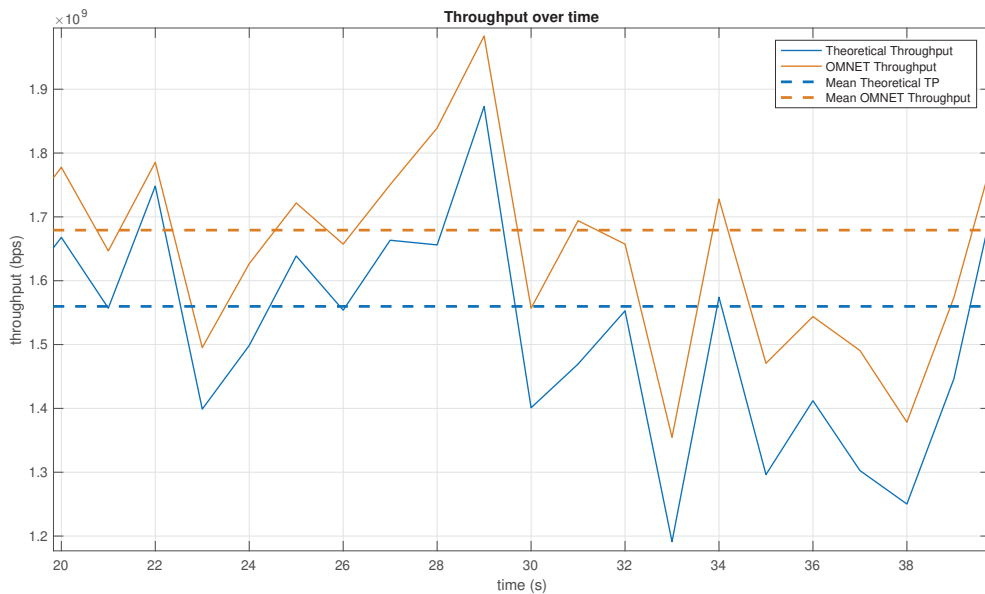
Figure 5.3: Plot of throughput over time for theoretical prediction and OMNET++ simulation; vectors B and B' used with an uplink vector of 1.5 Mbps – zoomed in.

which will increase the divergence between theoretical and real values. These are scenarios such as channel saturation and the delay caused by the uplink losses that are dismissed in the prediction. Although the arrays behave in a more distant manner as time passes, their mean is getting closer, as some of the assumptions in the theoretical prediction presume an infinite window of time for retransmissions, which can't be achieved by simulations of 100 s.

$$\delta = \frac{\left| \eta_{\text{theoretical}} - \eta_{\text{OMNET}} \right|}{\eta_{\text{theoretical}}} \times 100\% \tag{5.1}$$

The relative error calculated with equation 5.1 is 7.65 % and could be improved by a more complex algorithm implemented on *MATLAB*, and by running the simulations for longer periods of time in order to achieve "infinite" possibilities for retransmission.

5.3 Effects of Memory Delay on Throughput

After having analysed the maximum values achievable by the system, the memory access time was taken into account.

Table 5.10 shows the values of throughput and average transmission time when adding the memory delay to the simulations in *OMNET++*. For these simulations, the values for the best case scenario were omitted as it isn't influenced by this change, so the table presents the values with use of the power vectors B and B'.

Uplink data-rate	15 kbps	150 kbps	1.5 Mbps
Average Throughput [Gbps]	1.66392	1.63734	1.61685
Effective Throughput [Gbps]	0.26746	0.99945	1.25818
Average Trans. Time [μs]	16.8593	12.2863	11.0446

Table 5.10: Throughput performance with the effect of memory delays.

The delay degrades the throughput as predicted, being easier to identify the losses with the average throughput. This value is lower for higher uplink data-rates, as the feedback messages are received more often so the jumps in memory are more frequent. The effective throughput, while not as affected, is still lower by 5% for the best case (with uplink of 1.5 Mbps), and that is a problem that has to be mitigated.

Table 5.11 presents the values obtained for simulations where the algorithm was applied for an initial interval of 10.

Uplink data-rate	15 kbps	150 kbps	1.5 Mbps
Average Throughput [Gbps]	1.67031	1.63032	1.61689
Effective Throughput [Gbps]	0.46265	1.22410	1.26384
Average Trans. Time [μs]	15.7836	11.2729	11.0351

Table 5.11: Throughput performance with use of the algorithm for an interval of 10.

The improvement with this algorithm is clear, as though the average throughput isn't much affected, all the values for the effective throughput are higher. For the slowest uplink data-rate, the effective throughput almost doubles, and for the fastest, there is only a slight improvement under 1%. Optimisations have to be considered to improve this value.

The algorithm was then implemented for the optimal interval, i.e., 22 for small jumps and 43 for big jumps as presented in section 4.3.2. Table 5.12 presents the values obtained.

Uplink data-rate	15 kbps	150 kbps	1.5 Mbps
Average Throughput [Gbps]	1.67098	1.62893	1.61694
Effective Throughput [Gbps]	0.34650	1.16171	1.26581
Average Trans. Time [μs]	16.8593	11.3061	11.1148

Table 5.12: Throughput performance with use of the algorithm for optimal intervals.

For the fastest uplink data-rate of 1.5 Mbps, which is the case with more degradation from the memory delay, the effective throughput obtained with the algorithm is only 4% lower than the value obtained before introducing the memory delay factor, which is a high-bound to the maximum that could be achieved.

With this, it's possible to observe the benefits of the implementation of this algorithm which can be studied further to optimise its results.

Chapter 6

Conclusions

This chapter presents the conclusions obtained by this thesis and future work that can be done on its topic.

The topic addressed in this thesis was inserted in FSO communications, and aimed to overcome challenges present in this kind of transmission.

This thesis aimed to analyse the ARQ protocols that could be implemented in a LEO-to-ground system. Different protocol configurations were studied and implemented, and the performance of the achieved throughput was compared. The results of the simulations implemented were validated by theoretical formulae. Finally, the impact of memory delays was analysed and a mitigation solution proposed.

For the system with 1.5 Mbps of uplink data-rate, the fastest one studied, the implementation of an ARQ protocol allowed the effective throughput to improve from 0 Gbps to 1.31703 Gbps (for the channel's worst case scenario). This value was degraded with the consideration of the memory delay by 58.9 Mbps, which could then be improved by 7.63 Mbps.

The results were validated with a theoretical prediction and the simulations incurred an error of 7% which can generally be justified with the assumptions made.

The objectives for this thesis were achieved. The results obtained are relevant for the projects conducted in the Optical Communications Systems group at DLR, and the tools required for their implementation were delivered and can be used for simulations in the same area.

The presented thesis introduces an initial analysis of ARQ systems, and optimisations for the selected protocol. As a future work, the Optical Communications Systems group will continue the implementation of the proposed communication scheme, with consideration for the ARQ protocol recommended and the throughput values expected to achieve.

Prediction can be made on the average delay that each ARQ protocol arises, as well as studies through simulation on this criteria. Better understanding of the discrepancies on theoretical predictions and obtained values can be investigated on, with focus on the effect of the channel's coherence time.

As for the research in this topic, it would be interesting to test the system with different atmospheric events, such as scattered cloud coverage, and interaction with different ground stations. Furthermore, an adaptive-rate ARQ could be developed in order to overcome different channel conditions.

For future investigation concerning this subject, channel estimation can only be done posterior to receiving a message in full to calculate its error, which will be aggravated in a system with a slower data-rate on the uplink. To overcome this, machine learning and other prediction algorithms can be studied to guess the channel's behaviour and optimise the throughput of the system.

Bibliography

[1] M. K. Carson. *Alexander Graham Bell: Giving Voice to the World*. Sterling Publishing Company, Inc., 2007.

[2] M. A. Khalighi and M. Uysal. Survey on free space optical communication: A communication theory perspective. *IEEE Communications Surveys Tutorials*, 16(4):2231–2258, 2014. doi:10.1109/COMST.2014.2329501.

[3] H. Hemmati. *Near-earth laser communications*. CRC Press, 2009. ISBN:978-1-4200-1544-7.

[4] First high capacity space-to-ground laser communications system for the new european external iss platform bartolomeo. *Airbus*, 2018. URL:airbus.com/newsroom/press-releases/en/2018/03/first-high-capacity-space-to-ground-laser-communications-system-.htmlo.

[5] C. Schmidt, M. Brechtelsbauer, F. Rein, and C. Fuchs. OSIRIS payload for DLR's BiROS satellite. 2014. Optical Communications Group, Institute for Communications and Navigation, German Aerospace Center (DLR).

[6] ITU-T X.200, Information Technology - Open Systems Interconnection - Basic Reference Model: The basic model, 07 1994. International Telecommunications Union, URL:itu.int/rec/T-REC-X.200-199407-I.

[7] J. Proakis. *Digital Communications*. McGraw-Hill, 2001. ISBN:9780071181839.

[8] D. Giggenbach, F. Moll, C. Fuchs, T. Cola, and R. Mata Calvo. Space communications protocols for future optical satellite-downlinks. volume 4, 10 2011. 62nd International Astronautical Congress 2011, IAC 2011.

[9] L. C. Andrews and R. L. Phillips. *Laser Beam Propagation through Random Media*. SPIE Press, 1998. ISBN:0-8194-2787-X.

[10] B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics*. John Wiley and Sons, Inc., 1991. ISBN:9780471839651.

[11] A. Leon-Garcia and I. Widjaja. *Communication Networks: Fundamental Concepts and Key Architectures*, chapter 5.2. ARQ Protocols. McGraw-Hill School Education Group, 1st edition, 1999. ISBN:0070228396.

[12] S. Ahmadi. Chapter 7 - the IEEE 802.16m Medium Access Control Common Part Sub-layer (Part II). In S. Ahmadi, editor, *Mobile WiMAX*. Academic Press, 2011. ISBN:978-0-12-374964-2.

[13] S. Lin, D. J. Costello, and M. J. Miller. Automatic-repeat-request error-control schemes. *Communications Magazine, IEEE*, 22:5 – 17, 1985. doi: 10.1109/MCOM.1984.1091865.

[14] A. M. Michelson and A. H. Levesque. *Error-control techniques for digital communications*, chapter 6. Nonbinary BCH codes and Reed-Solomon codes. John Wiley and Sons, 1985. ISBN:0-471-88074-4.

[15] S. B. Wicker. Hybrid-ARQ Reed-Solomon coding in an adaptive rate system. In *IEEE International Conference on Communications, World Prosperity Through Communications,*, pages 1383–1387 vol.3, June 1989. doi:10.1109/ICC.1989.49907.

[16] H. D. Le, V. V. Mai, C. T. Nguyen, and A. T. Pham. Design and analysis of sliding window ARQ protocols with rate adaptation for burst transmission over FSO turbulence channels. *Journal of Optical Communications and Networking*, 11(5):151–163, 2019. doi:10.1364/JOCN.11.000151.

[17] V. V. Mai and A. T. Pham. Cross-layer designs and analysis of adaptive-rate transmission and ARQ for free-space optical communications. *IEEE Photonics Journal*, 8(1):1–15, 2016. ISSN.1943-0655.

[18] C. M. Schieler, B. S. Robinson, and D. M. Boroson. Data delivery performance of space-to-ground optical communication systems employing rate-constrained feedback protocols, 2017. doi:10.1117/12.2260669.

[19] M. Zorzi and R. R. Rao. Throughput of selective-repeat ARQ with time diversity in Markov channels with unreliable feedback. *Wireless Networks*, 2(1):63–75, 1996. doi:10.1007/BF01201462.

[20] ESA - types of orbits, 04 2017. European Space Agency, URL:esa.int/Our_Activities/Space_Transportation/Types_of_orbit.

[21] R. Perez. *Wireless Communications Design Handbook*. Elsevier, 1st edition, 1998. ISBN:978-0-12-550721-9.

[22] C. M. Schieler and B. S. Robinson. Data volume analysis of a 100+ Gb/s LEO-to-ground optical link with ARQ, 2018. doi:10.1117/12.2292910.

[23] B. S. Robinson, D. M. Boroson, C. M. Schieler, F. I. Khatri, O. Guldner, S. Constantine, T. Shih, J. W. Burnside, B. C. Bilyeu, F. Hakimi, A. G. an G. Allen, E. Clements, and D. M. Cornwell. Terabyte infrared delivery (TBIRD): a demonstration of large-volume direct-to-earth data transfer from low-earth orbit, 2018. doi:10.1117/12.2295023.

[24] M. Katzman. *Laser Satellite Communications*. Prentice Hall, 1987. ISBN:0-13-523804-8.

[25] CCSDS - Consultative Committee for Space Data Systems. Overview of space communications protocols, 2014. url:public.ccsds.org/Pubs/130x0g3.pdf.

Appendix A

Interface OMNET++

In the first screenshot of *OMNET++*, the editor is opened on the configuration file, where the user can easily edit the values for the data-rates, message lengths and simulation time. One can also easily select the power-vector to be used in the downlink and uplink channels separately.

In the second screenshot, a simulation is running. The interactive interface during the simulation allows the user to have a look at the actions occurring, which will simplify the debugging process.

Each module being simulated (i.e., the satellite, the ground station [*gndstation*], and the downlink and uplink clouds) can be separately opened, in order to see the log of each, timers set, arrays in use, and other features. The interface also allows to run the simulation in express mode until the end of the set limit time, or until a certain action, allowing some bugs in the code to be easily adjusted.

Simulation - version_Hybrid_06_6/omnetpp.ini - OMNeT++ IDE

File   Edit   Source   Navigate   Search   Project   Run   Window   Help

Quick Access   Simulation   Debug

Project Explorer

newerrortile.txt
not_AWGN_10kHz_100s.txt
OmnetCMDrun_AA.txt
omnetpp.ini
package.ned
perf.dat
real_channel_100s.txt
real_channel_13min.txt
real_channel_200s.txt
recip.dat
recip1.dat
satpacket.msg
some_downlink_vector.txt
SRARQ-FEC-BackwardErrorFreeChannel

Properties   Outline

version_Hybrid_06_6/omnetpp.ini

Resource

Property   Value
Info
  derived   false
  editable   true
  last modifie   September 12, 201
  linked   false
  location   /home/gonc_mf/[
  name   omnetpp.ini
  path   /version_Hybrid_0
  size   2,008 bytes

gndstation.cc   omnetpp.ini

Global Parameters

..MeanPowernWDOWN   = 3.2584E-06   # in Watts
..MeanPowernWUP   = 1.312199899E-08   # in Watts

..CodeWordLength   = 8*255   # "N" in bytes
..MessageLength   = 1779   # "K" in bytes
..sampling_frequency   = 10   # in KHz #Sampling rate

..downlink_data_rate   = 10e9   # bps
..uplink_data_rate   = 1.5e6   # bps

..CNACKCodeWordLength   = 255   # NACK FRAME Length in bytes
..CNACKMessageLength   = 217   # NACK MESSAGE Length in bytes

..simulationRuntime   = 100s   # Total Simulation Time in seconds - 13min=

downlink Parameters
**.downlink.downlink.ErrorCondition   = "file_input" # "errorfree" or "file_input"
**.downlink.downlink.ErrorFileName   = "B.txt" # KT04DL_10deg_Fs10kHz_1s # AWGN_10kHz_1s

uplink Parameters
**.uplink.uplink.ErrorCondition   = "file_input" # "errorfree" or "file_input"
**.uplink.uplink.ErrorFileName   = "B_up.txt"

Form   Source

Problems   Module Hierarchy   NED Parameters   NED Inheritance   Console   Search   Output Vector

CDT Build Console [version_Hybrid_06_6]

13:12:29 **** Incremental Build of configuration gcc-debug for project version_Hybrid_06_6 ****
make MODE=debug CONFIGNAME=gcc-debug all

13:12:29 Build Finished (took 435ms)

version_Hybrid_06_6/omnetpp.ini