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

Mafalda Gonçalves Lacerda Teixeira
mafalda.g.l.teixeira@gmail.com

Instituto Superior Técnico, Universidade de Lisboa, Portugal

October 2019

Abstract

The development of free-space optical communications (FSO) offers new challenges in the world of telecommunications. The possible configurations of an automatic repeat request (ARQ) to be implemented in a LEO-to-ground link were analysed. 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 and optimised for a higher throughput and lower channel saturation.
Keywords: Free-Space Optical Communications, Automatic Repeat Request, Cumulative ARQ, LEO downlink

1. Introduction

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.

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.

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.

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.

2. Background

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.

2.1. 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.

Channel Conditions

Free-space optical links convey different characteristics from RF channels [1]. 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 speed 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 speed 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:

$$P_{dBm} = 30 + 10 \log_{10} P_W \tag{1}$$

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.

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.

2.2. 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.

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 [3].

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 resends the packet. This scheme is limited in efficiency by the round trip delay time [3].

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 [3].

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) [3].

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.

2.3. 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) [4].

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.

The systems in place use Reed-Solomon (RS) codes, a specific type of Bose-Chaudhuri-Hocquenghem (BCH). 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 [5].

3. Conceptualisation

The three different protocols implemented varied on the type of feedback given to the space segment.

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.

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.

Negative Acknowledgements

The exchange is made by sending messages that report the missing or corrupted frames. Negative acknowledgements (NAK) operate with the supposition that the data is delivered properly and only re-sends data upon request.

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.

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.

3.1. Evaluation Criteria

The following paragraphs provide quantitative and qualitative criteria in order to compare the different protocols to be implemented and studied. These terms will be referred to when analysing the results obtained.

3.1.1 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 2 presents this, where η_{max} is the throughput high-bound, D_d is the data-rate of the downlink, and k is the number of information bits for n bits send, according to RS (k, n) =RS$(223, 255)$.

$$\eta_{max} = D_d * \frac{k}{n} \tag{2}$$

In order to have a consideration for the losses in the channel, one can induce the formula, where p and q are the probability of error in the downlink and uplink channels, respectively, and T is the 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.

$$T = 1*(1-p) + 2p*(1-p) + \\ +3p^2*(1-p) + 4p^3*(1-p) + ... \quad (3)$$

By grouping all of the cases in a sum, equation 4 is obtained.

$$T = \sum_{i=0}^{\infty} (i+1) \quad p^i*(1-p) \quad (4)$$

Considering this sum of infinite terms, its possible to induct it as being a geometric series expansion. With that, equation 5 is established.

$$T = \frac{1}{(1-p)^2}*(1-p) = \frac{1}{1-p} \quad (5)$$

Equation 5 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 6 deduces this, where u is the probability of receiving an uplink message.

$$u = (1-q) + q*(1-q) + q^2*(1-q) + \\ +q^3*(1-q) + ... \approx \frac{1}{1-q}*(1-q) = 1 \quad (6)$$

3.1.2 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, a generic time n time was considered and past times ($n-1h$, $n-2h$, ...) with a time-step h which is the round trip delay.

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.

$$R_n = 1*(1-p_n) + 2p_{n-1}*(1-p_n) + \\ 3*p_{n-2}*p_{n-1}*(1-p_n) + ... \quad (7)$$

The throughput is then calculated by dividing the data-rate and the ratio of the FEC by the average number of frames, presented by equation 8.

$$\eta_n = D_d * k/n * 1/R_n \quad (8)$$

3.1.3 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 1 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.

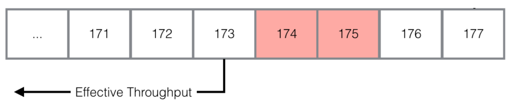

Figure 1: Example draft of the effective throughput.

3.1.4 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.

4. Implementation

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. 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.

4.1. 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 2.

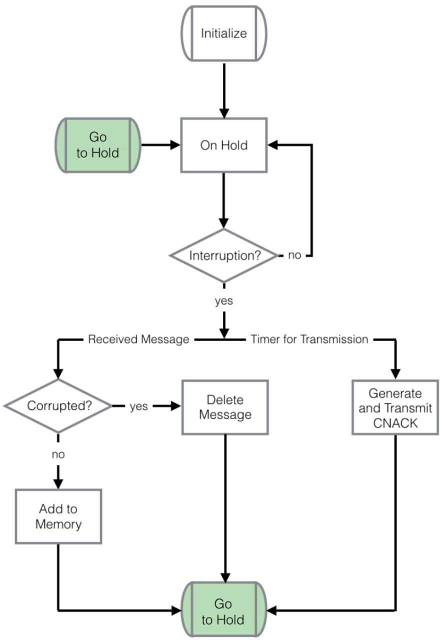

Figure 2: Simplified flowchart of the processing algorithm of the ground station.

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). 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 in order to interrupt itself and check the array and which cells are empty. Then it generates either a feedback frame and sends it.

4.2. Satellite

The flowchart in figure 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.

During retransmission state, the space terminal will send all the frames it knows that haven't been

[1]SN = sequence number

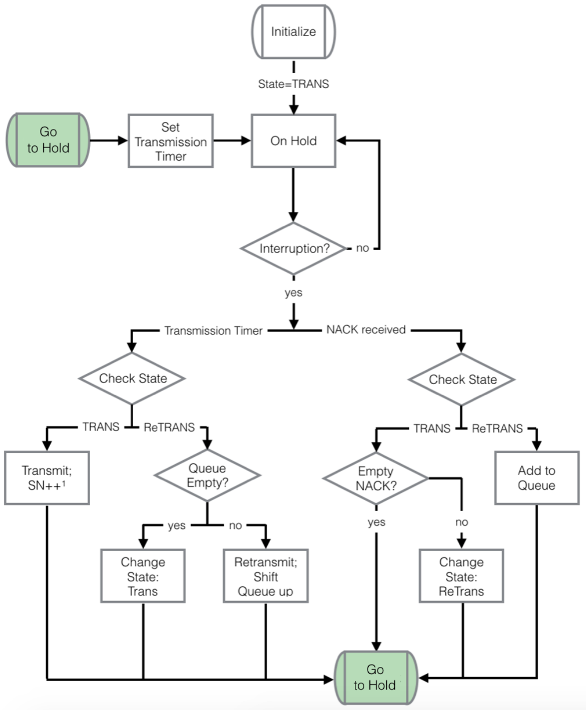

Figure 3: Simplified flowchart of the processing algorithm of the space terminal.

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.

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 delays as most transmissions can be queued up.

4.3. 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).

The different situations chosen were based on statistics for measurements at 5^o and 15^o elevations, which will correspond to the worst and best scenarios, respectively. These elevations were chosen, as only at 5^o link connection can be achieved, and above 15^o 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.

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. The message is corrupted by defining 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.

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.

Figures 4.3 and 4.3 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).

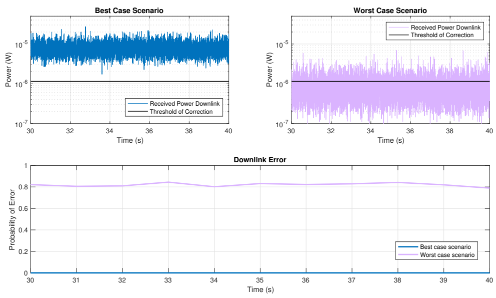

Figure 4: Power vectors for the downlink with and the probabilities of error, for the best and worst case scenarios (30–40s.)

As observable in the figures with representation of the received power and threshold of correction, the best case scenario has fewer instances below the threshold 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. 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.

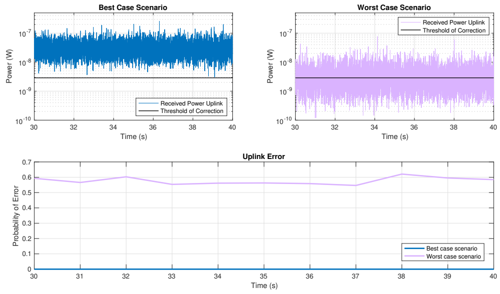

Figure 5: Power vectors for the uplink with and the probabilities of error, for the best and worst case scenarios (30–40s.)

4.4. 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 μs.

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.

4.4.1 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.

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, then the new cumulative NAK is added to the previous.

The algorithm has a variable input which is the value of the interval to be used. Figures 6 and 7 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 6 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 6: Example of a feedback packet with a "big" jump, applied in the one on the right.

Tables 7 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 7: Example of a feedback packet with a "small" jump, 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 (I_{opt}) for a maximum interval is, then, the number of transmissions ($M_d \div D_d$) that are sent during the same time as the saved memory delay (T_{mem}) – which is the time of one jump for the "small" case ($N = 1$) and of two in the "big" case ($N = 2$).

$$I_{opt} = \text{int}\left(\frac{T_{mem} * N * D_d}{M_d}\right) \quad (9)$$

Where int() refers to rounding the number up to the next larger integer. For the values used in the simulations, the optimum interval is 22 for the small jump and 43 for the big jump.

5. Results

Going back to equation 2, 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. Results for the Different Protocols

As an initial test, the three different protocols were tested. Tables 1 (best case scenario) and 2 (worst case scenario) show the values of throughput obtained for a simulation time of 100, downlink data-rate of 10 Gbps and uplink data-rate of 1.5 Mbps.

	No ARQ	ACK	NAK	Mixed
Avg.	8.72058	8.72058	8.72058	8.72058
Effect.	8.72058	8.72058	8.72058	8.72058

Table 1: Comparison of throughput in Gbps for different protocols for the best case scenario (15^o elevation).

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.

	No ARQ	ACK	NAK	Mixed
Avg.	1.68018	1.67995	1.67958	1.67966
Effect.	0	0.26533	1.31694	1.31703

Table 2: Comparison of throughput in Gbps for different protocols for the worst case scenario (5^o elevation).

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.

5.2. 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.

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 values are presented for the simulation that was run for 100 s, for a downlink data-rate of 10 Gbps and uplink data-rate of 15 kbps, 150 kbps, 1.5 Mbps, for the worst case scenario.

	15 kbps	150 kbps	1.5 Mbps
Avg.	1.68017	1.68011	1.67966
Effect.	0.15235	1.15658	1.31703

Table 3: Throughput in Gbps for different data-rates for the worst case scenario (5^o elevation).

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.

5.3. Validation of the Results

For the validation of the results, equations **??** and 8 were implemented in *MATLAB* with the algorithm presented in that section, and plotted alongside the vectors of throughput obtained from simulations. Figure 8 presents their plots over time, for the theoretical throughput, the throughput simulated in OMNET++ and the effective throughput, also from simulations.

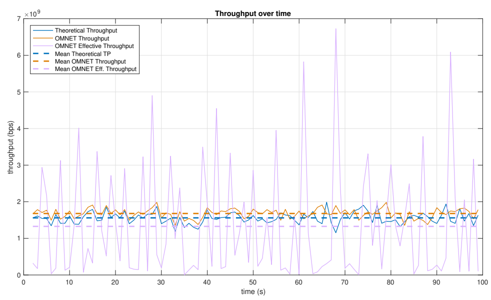

Figure 8: Plot of throughput over time for theoretical prediction and OMNET++ simulation (average and effective); 5^o elevation with an uplink vector of 1.5 Mbps.

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 low-bound 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 9 removes the effective throughput for clarity in the comparison of the average throughput (in theory and simulation) and figure 10 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.

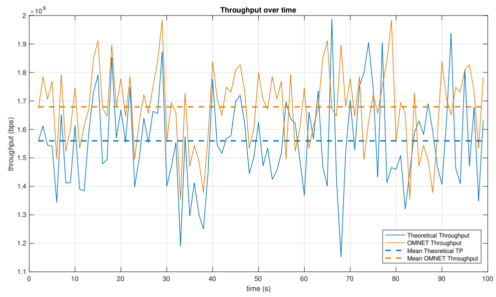

Figure 9: Plot of throughput over time for theoretical prediction and OMNET++ simulation; 5^o elevation with an uplink vector of 1.5 Mbps.

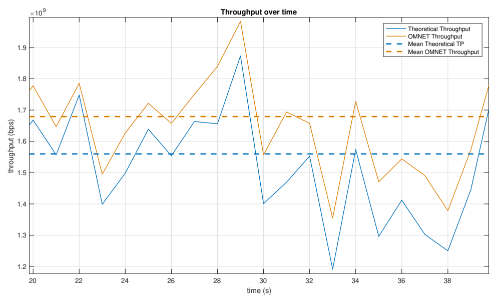

Figure 10: Plot of throughput over time for theoretical prediction and OMNET++ simulation; vectors B and B' used with an uplink vector of 1.5 Mbps – zoomed in.

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 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 100s.

The relative error obtained was 7.65 % and could be improved by a more complex theoretical algorithm, and by running the simulations for longer periods of time.

5.4. Effects of Memory Delay on Throughput

Table 4 shows the values of throughput and average transmission time when adding the memory delay to the simulations in *OMNET++* for the worst case scenario.

	15 kbps	150 kbps	1.5 Mbps
Avg.	1.66392	1.63734	1.61685
Effect.	0.26746	0.99945	1.25818

Table 4: Throughput in Gbps with the effect of memory delays for the worst case scenario (5^o elevation).

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).

The algorithm was implemented for the optimal interval, i.e., 22 for small jumps and 43 for big jumps. Table 5 presents the values obtained.

	15 kbps	150 kbps	1.5 Mbps
Avg.	1.67098	1.62893	1.61694
Effect.	0.34650	1.16171	1.26581

Table 5: Throughput in Gbps with use of the algorithm for optimal intervals.

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

6. Conclusions

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 datarate, 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.

Acknowledgements

The author would like to thank the Optical Communication Systems group at DLR and supervisors Jorge Pacheco and Fernando Nunes.

References

[1] 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.

[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] 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.

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

[5] 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.