Optimal Segment Length for Telecommand Data Link Layer Protocol
Work in Progress Report

C. Christelis and R. Wolhuter Department of Electronic Engineering
University of Stellenbosch
7600 Stellenbosch, South Africa
Email: christian@dsp.sun.ac.za, wolhuter@sun.ac.za

***Abstract*—**

This paper concerns itself with an investigation of optimising data throughput in space communications using the Telecommand data link layer protocol. The optimisation problem is seen from the perspective of the data link layer with payload being supplied by the next higher layer and the unit of measure used is effective data throughput defined as total payload received over total data transmitted. The optimisation is done by an on the fly segment length determination mechanism. The length is determined by the current FER which is estimated from the frame error history. Initial simulations have shown that effective data throughput may be vastly improved when using variable segment length as oppose to fixed segment lengths.

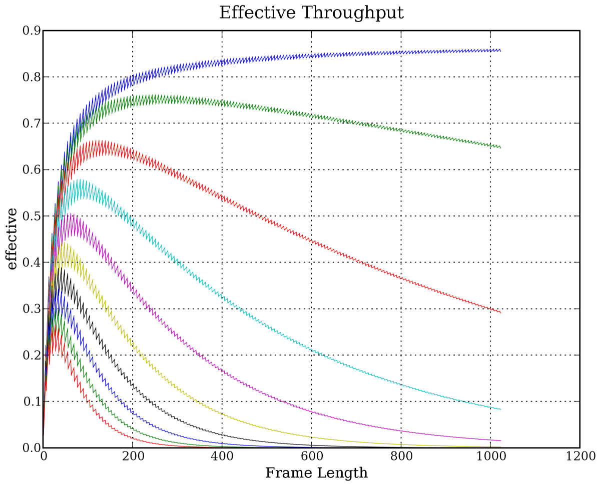
I. INTRODUCTION

The Telecommand data link layer (TC) protocol for space communications is an established standard in the space community. The TC protocol defines three sublayers within the data link layer each with a specific function. The three sublayers are as follows: The Segmentation sublayer which is responsible for splitting the payload data into segments, the Transfer sublayer responsible for reliability and the Synchronisation and Channel Coding sublayer which adds Forward Error Correction (FEC). [1] The redundancy added by the headers of the Segmentation and Transfer sublayer amount to 8 Bytes per segment. The data field to be inserted into a frame may be a variable length no greater than (2^{10} - 8) Bytes. The Segmentation and Channel Coding sublayer adds redundancy of 1 Byte of FEC for every 7 Bytes of Data. This sublayer also initializes the transmission by sending out a 2 Byte start sequence and it terminates with a 8 Byte stop sequence. Currently, a fixed length segment is recommended for a mission or a mission phase. In this paper it is briefly shown that choosing a fixed length packet size leads to a suboptimal data throughput and initial development of an on the fly optimal segment length calculator is proposed.

II. OPTIMAL FRAME LENGTH

The optimal frame length can be calculated for any given Bit Error Rate (BER). In [1]the following expressions for Frame Error Rate (FER) were given:

Fig. 1. Frame Length vs Effective Data Throughput



an uncorrected error in frame where$N = \lceil \frac{L+8}{7} \rceil$:
$P_{CY} = 1 - [(1-BER)^{63} + 63BER(1-BER)^{62}]^N$
a missed start sequence:
$P_{SY} = 1 - [(1-BER)^{16} + 16BER(1-BER)^{15}]$
a falsely accepted sequence
$P_{TY} = 1953BER^2(1-BER)^{61} + 651BER^3(1-BER)^{60}$
$FER = P_{TY} + (1-P_{TY})(P_{SY} + (1-P_{SY})P_{CY})$ (1)

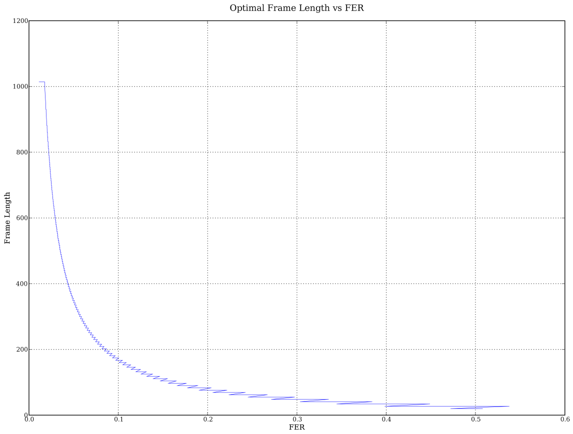
The effective data throughput, which is a function of both frame length L and BER, is given by:

$$eff = (1-FER)(\tfrac{7}{8})(\tfrac{L}{7\lceil \frac{L+8}{7} \rceil + 10}) \quad (2)$$

Effective payload data throughput versus payload data length for various BER's is given in Figure 1. The maximum points of these expressions were found by the aid of a Python program, but since the BER can never be known by the Segmentation sublayer without compromising the independence of each sublayer, an estimate needs to be made. The maximum point of each BER was found and the FER compared to the Optimal Frame Length (L$_{opt}$). This gave rise to the results shown in Figure 2.

The optimal segment length for a given FER can be approximated very closely by the expression given in (3). The fact that the results obtained from this analysis can be approximated by

Fig. 2. Optimal Frame Length vs FER



a simple expression is convenient, suffering evaluation without much computational effort.

$$L_{opt} = \lceil (\frac{18-16 \cdot FER}{7 \cdot FER}) \rceil \cdot 7 - 1 \quad (3)$$

III. ESTIMATING FER FROM FRAME ERROR HISTORY

In the previous section it has been shown that a good estimate for the optimal payload, length given the precise FER, can be made by (3). This section investigates methods of obtaining a FER estimate ($\hat{FER}$). The $\hat{FER}$ must be made from a history of samples. This history has been assigned two characteristics which will later be used later in scenario planning. The first is the history depth h. This characteristic simply defines the number of samples to be used in the estimation of the FER. The second is a weighting of samples w. This characteristic assigns a weight to the samples which lets more recent samples have a higher weight in determining the $\hat{FER}$. This might make sense in conditions where the BER changes sharply. Four values for h have been chosen: h = 50, 100, 200 and 400. These values are based on confidence interval calculations [2]. The results of these calculations are summarised in table I.

TABLE I
95% CONFIDENCE INTERVALS AT FER

	FER = 0.01	FER = 0.1	FER = 0.5
1%	h = 380	h = 3457	h = 9604
5%	h = 15	h = 138	h = 384
10%	h = 4	h = 35	h = 96

Choices for the values of w have not yet been made beyond $w = 1$, but will be constrained by the maximum weight of a single reading.

IV. SIMULATION AND SCENARIO PLANNING

A simulator has been designed and programmed in C++. This simulator is based on a skeleton simulator designed and written by the University of Stellenbosch Computer Science Department. The simulation simulates the flow of packets through the TC data link layer. A single unit of measure for each simulation run was chosen. This is the payload data received over total data transmitted, referred to as effective data throughput.

Scenarios in which the various h and w combinations are to be tested include various constant BER's, increasing BER's and decreasing BER's. A full scenario plan will be the result of further work [3].

V. INITIAL RESULTS

The simulation has been run with the history depth h values as chosen above without any weighting ($w = 1$) and constant BER. The effective data throughput, for 100000 successfully transmitted packets, was obtained and the simulation repeated 1000 times with a unique seed for each run. The results of these initial test runs are given in table II.

TABLE II
INITIAL SIMULATION RESULTS

		BER = 0.00025		BER = 0.001		BER = 0.004		BER = 0.008	
w	h	Mean	Dev.	Mean	Dev.	Mean	Dev.	Mean	Dev.
1	50	84.1938	.0379	75.4159	.0674	47.8165	.1123	19.4434	.0246
1	100	84.3676	.0362	76.0547	.0661	48.2067	.1190	19.4439	.0246
1	200	84.4785	.0355	76.2513	.0663	48.3947	.1182	19.4450	.0246
1	400	84.5465	.0358	76.3289	.0677	48.4863	.1210	19.4473	.0247

The segment length required to achieve 84 % effective throughput at BER = 0.00025 is L = 1014. L = 1014 yields a theoretical effective throughput of $<$ 2 % at BER = 0.004 and virtually 0 at BER = 0.008.

VI. CONCLUSION

The initial results show, that the proposed mechanism of on the fly optimal segment length calculation may indeed yield favourable results. The effective throughput of payload data with variable segment length has, over the simulated data range, outperformed constant segment lengths. Work is still to be done in the areas of Scenario Planning, Simulation and Statistical Analysis.

REFERENCES

[1] ESA Requirements and Standards Division, "ECSS-E-50-04A," ECSS, Tech. Rep., 14 November 2007.
[2] "Network performance modelling course," February 2008.
[3] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed. McGraw-Hill, 2000.

Christian Christelis Is currently a masters student at the University of Stellenbosch.

Riaan Wolhuter Is a senior researcher of the DSP Telecomms group at the Faculty of Engineering at Stellenbosch University, South Africa.