GEO Relay Applications and the Usage of Reed-Solomon Codes in such systems

M.L. Pors

ET4030 - Error Correcting Codes
Delft University of Technology
M.L.Pors@student.tudelft.nl

January 5, 2015

1 Introduction

In [1] a satellite network with a GEO (Geostationary Earth Orbit) data relay satellite and LEO (Low Earth Orbit) satellites is considered, where the writers look into the challenges that occur implementing such a system. In the considered system, maximum independence between multiple stages is achieved by packetizing all data in these different stages and both the laser and microwave links are equipped with several individual error-control coding schemes, such as Reed-Solomon Codes. It follows that even with the usage of well-known traditional error-control coding schemes in LEO and GEO, the interaction between coding, framing and reframing operations and the relations between various packet error rates is challenging but essential for performance assessment.

In this essay I will review this research. I will discuss the settings of relay satellite networks, give an overview of the data processing scheme used in the system of [1] and elaborate on the used error-control codes used here. This elaboration will contain a discussion of a (255,239) Reed-Solomon (RS) Code, which is a cornerstone of the system. An experiment is performed to investigate the notion of 'RS on RS' ([1]).

For background information on Error-Correcting Codes, I relied on [2] and [6]. More information on simulating codes such as RS, I relied on [3] and [4].

2 GEO Relay Applications

An abstract overview of a GEO relay application with LEO satellites is given in Fig. 1. LEO satellites as used for earth observation purposes are producing typically high amounts of data. To get this data to the ground station (GS), the data from the LEO satellite is sent via a GEO relay satellite. Before I will review the processing steps of each part of the system (Section 3), the system itself and the benefits of using a GEO data relay is discussed.

Information transmission in a GEO relay satellite network is divided into two steps. First, the connection between the LEO and GEO can be an Optical Inter-Satellite Link (OISL), where the LEO satellite uses a laser beam to send the data to the GEO. An OISL has several benefits: Availability of high data rates (up to 1800 Mbps) and robustness against interference from other systems as well as interference to other systems or antennas on the same platform.

In the second step, data from the GEO to the GS is transmitted over a downlink (DL). In particular this transmission has several benefits over a direct transmission from a LEO satellite to the GS. Because the GEO has a stationary position above the Earth, it doesn't need a steerable antenna to send to the GS, as opposed to the LEO which does need one. Also, using the GEO relay increases the available transmission length. LEO satellites are characterized by intermittent connectivity with their ground stations. Contacts are short and separated by long intervals [1, 5]. When the LEO has a steerable antenna, a GEO is in its beamwidth for halve an orbit length, a considerable longer period. This results in an enlarged time window, which also increases the data rate instantaneously. Another benefit is that one GEO relay can serve several LEO satellites in sequence (but not in parallel).

Both the DL between the GEO and GS and a direct optical link between the LEO and GS are vulnerable to interference. A direct optical link will become unavailable due to clouds, where the GEO and GS link can endure. Here, the received optical signal is split in the GEO relay into N data streams transmitted on N microwave downlinks. Here, the Ka-band is used (27-40 GHz).

Examples

In [1], two examples of the GEO relay applications of Fig. 1 are discussed shortly. A future application will be the EDRS (European Data Relay Satellite) system in context of GMES (Global Monitoring for Environment and Safety) Beside this, LEO satellites can be replaces by a HAP (High Altitude Platform) or UAV (Unmanned Aerial Vehicle). In [5] the DTN (Delay-/Disruption- Tolerant Networking) architecture is discussed, which also uses GEO relays.

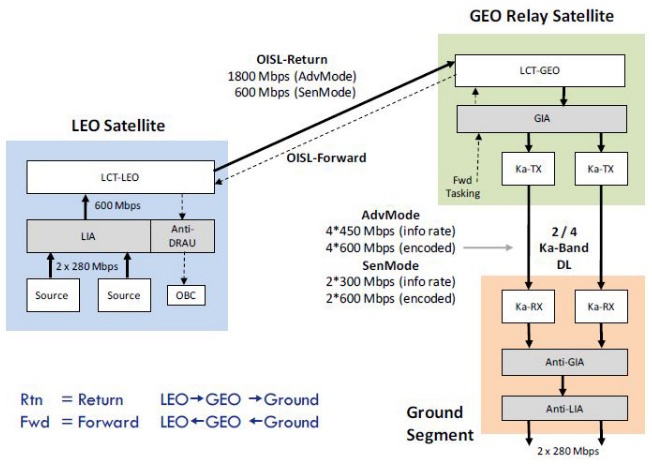

Figure 1: GEO relay application [1]

3 Data Handling

The data handling schemes are implemented in the LIA and GIA (LEO and GEO Interface Adapter), as indicated in Fig. 1. Framing and coding takes place before the data is sent to the next segment. At the GS the inverse operations of those in the LIA and GIA are performed in the Anti-GIA and Anti-LIA. From Fig. 1, one can directly see the order of the operations will be such that there is encapsulation (i.e. the Anti-GIA comes before the Anti-LIA segment).

Date rates

An important notion in the data handling scheme that follow from Fig. 1 and Fig. 2 is the adaption of the data rate such that it 'fits' the used channels. Namely, the data rate of the Mass Memory (MM) of the LEO is smaller than the date rate of the OISL and DL.

The data rate of the MM, which is 280 Mbps, can be seen as the source data rate. Because the laser connection can not be easily be adapted, but the amount of data fed to the channel can, in the LIA, data from multiple sources are taken and multiplexed after the data processing steps. Idle frames with fixed content are added to achieve the correct channel data rate, 600 Mbps. Using these idle frames to reach this channel data rate is important because then the system can work with the same 'clock' at the LIA, GIA and GS segment. This ensures that the data from the LEO reaches the GS synchronously and that there isn't the possibility of data to arise out of the blue.

Also important is that although the frames after encoding in GIA have an identical length in bytes as the LIA frames (clear from below), the GIA frames are shorter in time due to a higher data rate.

Data Processing

A detailed overview of the data processing steps in the satellite network can be seen in Fig. 2. Here also the error-correcting codes capabilities are shown. One who is interested in the precise data processing steps should advise [1].

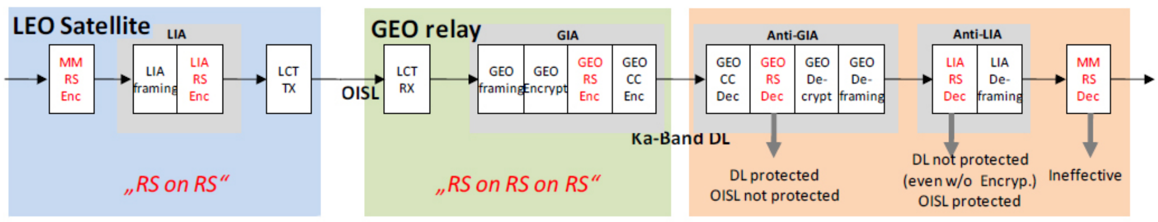

Figure 2: Capabilities of the error-correcting codes in the satellite network [1]

As can be seen in Fig. 2, the most important data processing steps are framing and en-/decoding. The error-correcting code that is used is a (255,239) Reed-Solomon Code (discussed in Section 4.1). In the LIA, blocks of 1912 bytes are created (including a 2-byte header) on which this RS code is performed 8 times: $8 \times 239 = 1912$. The result of $8 \times 255 = 2040$ bytes gets an additional 4-bytes ASM, resulting in a frame length of 2044 bytes. The channel data rate of a LIA frame is therefore $280 \times 2044/1910 = 299.644$ Mbps. The data can be transmitted over the OISL with a channel data rate of 600 Mbps, which is achieved by multiplexing two source frames and a small amount of idle frames with an average data rate of 0.712 Mbps.

The processing in the GIA follows a similar pattern. Using the data from the LIA where the LIA frame structure is ignored, again blocks of 1912 bytes are constructed such that the RS code can be used. Also here the result block of 2040 bytes gets a 4-bytes ASM such that it is ready for transmission. From Fig. 1 it can be seen that the DL has two different modes for transmission. A convolutional code is applied with the code rate of 2/3 and 5/6 to achieve the correct data rate. Also here, idle frames are needed to fill the gap to 600 Mbps.

At the GS the reversed operations take place, such that the original data from the LEO MM is retrieved. The GEO encapsulates the data from the LEO, such that GEO decoding and reframing must take place first before we have the structure from the LIA and the decoding of these frames can take place.

4 Reed Solomon Codes

From Section 3 it is clear that a (255,239) Reed-Solomon code is used multiple times to encode the data. Here, we elaborate on Reed-Solomon and the notion of ‘RS on RS’, which occurs in Fig. 2.

4.1 Introduction to RS Codes

Reed-Solomon Codes are linear cyclic block codes, a subset of the BCH codes called non-binary codes [2, 3]. Of most interest are the RS codes over GF(2^m), because most applications such as the satellite network make use of binary data. RS codes are particular useful against burst errors and have been used in hard disks for the past 15 to 20 years. RS codes are also useful for channels having memory like CD and DVD. Other popular applications are deep space and satellite, microwave and mobile communications. Given this and the fact that interference due to clouds result in burst errors, it follows that RS is a good choice for the GEO relay application.

Properties of the (255, 239) RS Code

Like BCH codes, RS codes are characterized by two parameters: m determines the lenght of the code and t determines the code's Hamming distance or symbol error correcting capability [2]. The $(n,\, k)$ RS code contains k data symbols and $n-k$ parity symbols, which do not ‘contain’ data, but are used to calculate which bits are in error (if so). The relation between length n, dimension k and m and t is:

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) \tag{1}$$

From this, it follows that for the (255, 239) RS code, $m = 8$ and $t = 8$. The alphabet for the code is thus GF(2^8). This is a popular choice for may applications because GF(2^8) = GF(256), where each of the 256 field elements can be represented as a byte (8 bits) [6]. The distance between two codewords $d = 2 \times 8 + 1 = 17$. This RS code can correct burst errors up to a length of $t = 8$ bytes.

4.2 RS on RS

In Fig. 2, the notion of 'RS on RS (on RS)' is made. RS encoding is done multiple times over different blocks of data. This 'RS on RS' is not a concatenation of block codes in the usual sense. Namely, while after MM and LIA both have a length of 2044 bytes, the information word at the input of the coding sections is smaller. In other words, one MM frame is encapsulated by two LIA frames at least and three LIA frames at most [1].

At the GIA a similar thing happens, as can be seen in Fig. 3. GEO frames are constructed from the LIA frames and after an encryption with a CBC block cipher and addition of an 8-bytes header, RS is applied. Coding capabilities are given in Fig. 2. The GIA RS code is used for the protection of the DL and errors on the OISL are invisible for the GIA. The writers of [1] state that the LIA RS can not protect both the OISL and DL due to the following reason: "If errors caused by the DL are remaining after the GIA decoder, then the error structures are such that the LIA RS decoder correction capability will be typically exceeded".

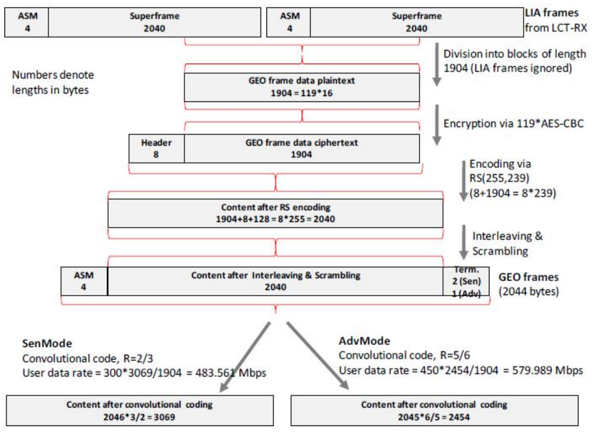
ASM 4
Superframe 2040
ASM 4
Superframe 2040
LIA frames from LCT-RX
Division into blocks of length 1904 (LIA frames ignored)
Numbers denote lengths in bytes
GEO frame data plaintext 1904 = 119*16
Encryption via 119*AES-CBC
Header 8
GEO frame data ciphertext 1904
Encoding via RS(255,239) (8+1904 = 8*239)
Content after RS encoding 1904+8+128 = 8*255 = 2040
Interleaving & Scrambling
ASM 4
Content after Interleaving & Scrambling 2040
Term. 2 (Sen) 1 (Adv)
GEO frames (2044 bytes)
SenMode
Convolutional code, R=2/3
User data rate = 300*3069/1904 = 483.561 Mbps
AdvMode
Convolutional code, R=5/6
User data rate = 450*2454/1904 = 579.989 Mbps
Content after convolutional coding 2046*3/2 = 3069
Content after convolutional coding 2045*6/5 = 2454

Figure 3: Framing operations in GIA [1]

Simulating RS on RS

To investigate this property, I created a small example using fuctions explained in [3, 4]. It should be so that if the decoding of a GIA frame 'fails', the decoding result of the LIA frames where this frame is constructed of should also contain errors. Using the MATLAB code from Appendix A, I investigated the consequences of a burst error in a GIA frame on the LIA frames it is constructed from. We start with three LIA frames of 239 bytes which are used to create two GIA frames, after they are encoded using a (255,239) RS code. After encoding the GIA frames using the same code, noise is added to the second GIA frame, which is constructed from the end of the second and the beginning of the third LIA frame. The next steps are the decoding steps and checks whether we get the original GIA frame and LIA frames.

The most important lines of code are line 35 and 37. Here the length and the position of the burst error in the GIA frame are defined. Adapting these lines to investigate multiple situations, I made the following observations:

- When the burst error is of length ≤ 8, the GIA frame is correctly decoded, as well as the LIA frames. An expected observation.
- When a burst error of length > 8 bytes is totally located at one of the LIA frames (as in Appendix A), this frame can not be decoded without any errors. The other LIA frame will be decoded correctly, assuming that the other GIA frame that contains the rest of the LIA frame contains no errors.
- When a burst error of length > 8 is located at the 'border' of the two LIA frames, there are possibilities that the GIA frame is decoded with errors, but the LIA frames are decoded without errors. In this case, a LIA frame will be decoded without errors when the lenght of the error located in the LIA frame after the GIA decoding is ≤ 8 bytes.

Simulation Conclusions

Given the results above, we can look what consequences this has on the GEO relay application. Note that in this example, the CBC encryption step is skipped and that the RS code is only performed on 239 bytes and not multiple times on a larger frame on 1912 bytes. This should not have any influence on the function of the RS code itself, and thus the result of this experiment.

Given $t = 8$, a decoded GIA frame (1912 bytes) will contain errors if there is minimal one burst error of length > 9 in one of the encoded segments of length 255 bytes. Assume that this burst error (length $= 9$) occurs at the segment that contains the border between two LIA frames. The start of this burst error can occur on $255 - 9 = 246$ locations. A incorrect decoding of this GIA frame but correct decoding of the LIA frames occur when the burst error is crossing this border. Illustrated in Fig. 4, there are 8 start positions of the burst error that assure this (from 8 bytes before the border to 1 byte before the border). Thus, the possibility that the burst error is located on a position that leads to a correct decoding of the LIA frames is $8/246$. It is clear that this is a small possibility, but this gets even smaller because a long burst error in one of the other segments which do not contain the LIA frame border will always lead to a incorrect LIA frame decoding.

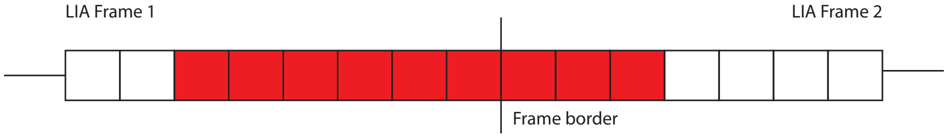

Figure 4: A burst error of length 9 crossing the LIA frame border

From above it follows that it is correct to say that if errors caused by the DL are remaining after the GIA decoder, then the error structures are such that the LIA RS decoder correction capability will be typically exceeded. This means indeed the DL is not protected by the LIA RS. [1] states that this situation could be changed if a deep interleaving over several frames is applied at the GIA input.

Using a similar argument, the MM RS code is more or less inefficient, both with regards to OISL or DL errors, however, for error detection on the user layer it could be very beneficial.

5 Review of Selected Codes

Although this report mainly focus on the GIA Relay Systems itself and the use of RS codes in this system, it is interesting to name some of the observations made in [1] regarding the performance of the selected codes.

Fist, the writers agree that the selected well-known RS code with interleaving over 8 RS words is not the optimum coding scheme. The error structure of the OISL is characterized by independent bit errors with constant error rate per GIA frame, while (as clear from Section 4.1) RS codes are particularly strong against burst errors. On the other hand, there were no overwhelming improvements found.

Second, Low-Density-Parity-Check (LDPC) codes were tried as alternative. Using these codes, an improvement in the order of 1.5 dB could be reached. One who is interested could examine in which way LDPC could be applied in the system - [1] did not elaborate on this solution. LDPC was not considered attractive after considering the trade-off between coding gain, link budget requirements, heritage and processing effort.

Third, an alternative scheme for interleaving can be superior to the convential way of interleaving. Interleaving is the process where the symbol sequence is rearranged in a deterministic way [2]. For example, codewords can be collected row-wise, but transmitted column-wise, in which way burst errors can be spread over different codewords. For the system in [1], the data can be read into the interleaver matrix memory column-by-column (conventional) scheme or row-by-row (alternative). In the second manner, the header is contained only in the first codeword, which reduces the Frame Header Error Rate. Also, errors due to enciphering and decryption have less devastating effects on the number of remaining errors.

6 Conclusion

In this report, a satellite network with a GEO data relay satellite and LEO satellites is considered. In this network, data is sent from the LEO satellite via the GEO data relay towards a groundstation (GS). To get a good understanding of the system and its challenges, there is elaborated on the setting and processing steps in all three segments, the LIA, GIA and GS.

The error-correcting code that forms a cornerstone of the system is the (255,239) Reed-Solomon Code. In the system, the notion of "RS on RS" is done. With an experiment I investigated the consequences of a burst error in a GIA frame on the LIA frames it is constructed from. It followed that, as stated in [1], it is true that if errors caused by the DL are remaining after the GIA decoder, then the error structures are such that the LIA RS decoder correction capability will be typically exceeded.

One challenge that has not come forward yet is that, since LEO satellites have an operational lifetime of typically 5 years compared to typically 15 years for GEO satellites, it is essential that technologies implemented on GEO do not restrict the development of new innovations for LEO satellites. An error-control encoding on LEO shall be terminated by decoding only at GS but not at GEO. This is supported by performing encapsulation and reframing operations at several instances of the satellite network.

This report was written as part of the course Error-Correcting Codes, where [1] was given as the basis. Although I probably could have elaborated more on the different used codes, reading this article I was first determined to fully understand the processing steps of the system. This partly explains the somehow detailed summary of the GEO Relay Application. Also, in understanding of the system, I found that much of my knowledge of other courses was very useful. As a student that has been focusing on electrical engineering topics for a short period, it was a nice experience to put the new knowledge into perspective. Given that this was one of the goals for this report, I was very pleased.

Secondly, I was directly interested in the notion of "RS on RS" and wanted to do some experimenting with this, devoted to check whether the statement of the writers was really true. As noted in Section 5, one could also investigate in which way a LDPC code can improve the well-known RS code, but I found this less interesting and the article gave less information about this for a good start. Where I think I can still gain some more in the detailed understanding of Reed-Solomon Codes (and others), it was informative to try to use these codes. It also motivated me to stay interested in the topic of error-correcting codes.

References

[1] B. Friedrichs and P. Wertz, "Error-control coding and packet processing for boradband relay satellite networks with optical and microwave links," *6th Advanced Satellite Multimedia Systems Conference*, 2012.

[2] J. Weber, *Lecture Notes on Error-Correcting Codes*, 2013.

[3] M. Viswanathan, *Simulation of Digital Communication Systems using MATLAB*, 2013.

[4] MathWorks, *Documentation on Reed-Solomon Codes for MATLAB.* [Online]. Available: nl.mathworks.com/help/comm/reed-solomon-codes.html

[5] P. Apollonio, C. Caini, and M. Lulf, "DTN LEO satellite communications through ground stations and GEO relays," *PSATS 2013, LNICST 123*, 2013.

[6] B. Sklar and F. J. Harris, "The ABCs of linear block codes," *IEEE Signal Processing Magazine*, 2004.

A Simulating RS on RS: MATLAB Code

```matlab
% From 3 LIA frames, 2 GIA frames are created.
% This script finds out what the size of the burst error in the second GIA
% frame has for consequences on the second and third LIA frame.
clear all
m = 8;
t = 8;
n = 255;                        % 2^m − 1, thus follows m = 8;
k = 239;                        % 2^m − 1 − 2t, thus follows t = 8;
hEnc = comm.RSEncoder(n,k);     % Creation of encoder
hDec = comm.RSDecoder(n,k);     % Creation of decoder

%% Construction of three LIA words of size 239
lia1 = randi([0 n], k, 1);
lia2 = randi([0 n], k, 1);
lia3 = randi([0 n], k, 1);

% RS−encoding of the three LIA words, resulting in size 255.
liaenc1 = step(hEnc, lia1);
liaenc2 = step(hEnc, lia2);
liaenc3 = step(hEnc, lia3);

% Construction of two GIA words of size 239;
% GIA1 contains the last 84 symbols of LIA1 and first 155 of LIA2
% GIA2 contains the rest of LIA2, filled with 139 sybols of LIA3
gia1(1:84, 1) = liaenc1(172:255); gia1(85:239,1) = liaenc2(1:155);
gia2(1:100, 1) = liaenc2(156:255); gia2(101:239,1) = liaenc3(1:139);

% Encoding of the GIA words, resulting in size 255.
giaenc1 = step(hEnc, gia1);
giaenc2 = step(hEnc, gia2);

%% Adding noise to the second GIA word.
% It should be that a the amount of errors ≤ 8 bytes can be corrected.

err = 12;                   % length of burst error in the beginning of GIA2.
noise(1:255,1) = 0;                             % Constructing the error vector
noise(4:(4+err−1),1) = (1+randi([0 n−1], 1, err));       % Actual errors
gianoisy = gf(giaenc2,m) + noise;    % Creation of GIA2 with burst error.

%% Decoding of the GIA words.
[giadec1, nerrse1] = step(hDec, giaenc1);
[giadec2, nerrse2] = step(hDec, double(gianoisy.x));

% Check whether decoding of GIA2 worked.
isequal(giadec2, gia2)

% LIA2 is constructed from the decoded GIA1 and GIA2
newliaenc2(1:155,1) = giadec1(85:239);
newliaenc2(156:255,1) = giadec2(1:100);
[nlia2, nerrsnlia2] = step(hDec, newliaenc2);

% Check whether decoding of LIA2 worked;
% This should only be true if the other check is also true.
isequal(nlia2, lia2)

%% Addition to check whether LIA3 stays intact
% The error occured at the LIA2 part of GIA2, LIA3 should be is still OK.
% We construct a new encoded LIA3 using the decoded part of GIA2 and the
% rest of original encoded LIA3.
newliaenc3(1:139,1) = giadec2(101:239,1);
newliaenc3(140:255,1) = liaenc3(140:255,1);
[nlia3, nerrslia3] = step(hDec, newliaenc3);
isequal(nlia3, lia3)
```