

# Efficiency Comparison of LDPC-LDGM and Raptor Codes for PL-FEC with Very Large Block Sizes

Rafael Montalbán Gutiérrez and Gonzalo Seco-Granados  
*Signal Processing for Communications and Navigation Group (SPCOMNAV)*  
*Universitat Autònoma de Barcelona (UAB)*  
*Edifici Q - Campus UAB, 08193 Bellaterra (Barcelona), Spain*  
*Phone: +34 93 581 35 62, Fax: +34 93 581 40 31*  
*e-mail: rafael.montalban@campus.uab.es, gonzalo.seco@uab.es*

## Abstract

*In this paper the performance of LDPC-LDGM codes and Raptor codes is studied and compared in terms of efficiency. LDPC-LDGM and Raptor are FEC codifications that can be applied at packet level. PL-FEC consists in including redundant packets in the transmission to ensure correct reception. Although these techniques have been previously studied in the literature, they have usually been compared with the performance of small block Reed Solomon codes. This paper presents a novel comparison between the performance of LDPC-LDGM codes and Raptor codes as alternatives for packet level FEC when large codification blocks are needed. The results obtained show that Raptor codes clearly outperform LDPC-LDGM codes in almost all scenarios.*

## 1. Introduction

Nowadays, packet-level forward-error correction (PL-FEC) codifications are applied to many systems, from standards designed for mobile terrestrial TV, such as ETSI's DVB-H [1], to satellite communications based e.g. on DVB-SH/S-2 standards. Packet level FEC helps to ensure the correct reception of data, reducing or avoiding the loss of packets caused by blockages and signal fading.

A paradigm of systems where PL-FEC is pivotal is the modification of the wideband fixed satellite communication standard, DVB-S2/RCS, to encompass mobile applications [2]. For many years, Ku band has

been used with great commercial success to supply multimedia contents to multiple fixed terminals. This kind of systems allow us to provide service to a potentially limitless number of users, as long as they are in the satellite coverage area. Moreover, these systems have also been used to grant users Internet access, encapsulating IP packets inside satellite standards Packet Data Units (e.g. the PDU defined by ETSI's DVB-S/S2). It was only a matter of time that extending content provision to mobile users was subject of study. The large coverage offered by satellites, makes the idea very promising. One of its problems is that mobile terminals have important power and size limitations. However, the use of satellite systems to provide services to collective users sharing the same infrastructure, such as trains, ships, buses, planes... has proved to be feasible.

This new use of satellites raises new problems as standards used nowadays were designed for transmissions to fixed terminals and are not prepared to manage the mobile channel conditions. We can see an example of this type of satellite-land mobile channel in the railroad scenario in [2]. The problems faced in the mobile channel imply that new standards or modifications to those existing will be needed to guarantee quality of service and data integrity. It is also important to note that these modifications should be preferably done in the upper layers of the standards in order to take advantage of the existing infrastructure.

Specifically, fading and shadowing phenomena may cause partial or total loss of the data being transferred. These phenomena can last for a period of time comprising several packets [2], [3] and therefore, we need a way to ensure correct reception of sent packets. Traditionally, in point to point communications, ARQ

---

*This work has been partially supported by project TEC2008-06305 and scholarship COLAB2008.*

(Automatic Repeat reQuest) is used, besides obviously error-correction at bit/symbol level. When a packet is not received, a resend is asked for. However, in our scenario ARQ is not suitable as we face very high return times and a limited return channel capacity. Moreover, ARQ is not suitable at all in broadcasting, because we do not normally have a return channel, or when it is available, ARQ would imply frequent repetitions of data for one or other of the terminals. Other techniques suitable for broadcast systems are diversity techniques, both in time and space [3], [4], and packet-level forward-error correction.

In this paper, we focus on large block-FEC codes suitable for packet-level codification. Precisely, this paper presents novel results comparing LDPC (Low Density Parity Check) LDGM (Low Density Generator Matrix) codes and Raptor codes. Both of them, individually and in comparison to Reed Solomon codes, have been subject of previous investigations. In [5], LDPC-LDGM and its variants are analyzed in-depth and compared to a Reed-Solomon code. Likewise, Raptor code's performance has been compared to Reed Solomon in simulations by the 3GPP (3rd Generation Partnership Project) and in papers such as [6]. Here, for the first time to the best of the authors' knowledge, LDPC-LDGM codes and Raptor codes are compared to each other in terms of codification efficiency.

In Section 2, we briefly introduce the PL-FEC concept and some FEC codifications applicable to large amounts of data. Section 3 is specifically dedicated to the comparison between Digital Fountain's Raptor codes and the LDPC-variant LDGM codes developed by INRIA (Institut National de Recherche en Informatique et en Automatique), and selected simulation results are shown. Finally, in Section 4 we draw the conclusions.

## 2. Packet Level FEC Techniques

The complex mathematical fundamentals of the different codifications are beyond the scope of this paper. However, a brief explanation of FEC codification basic concepts is needed in order to understand the differences between packet level FEC and bit level FEC, as well as the results presented in the next section. Moreover, a basic explanation of the operation of LDPC LDGM and Raptor codes will be presented. In the following explanation, the terms code and codification are used indistinctly.

Generally, a FEC codification consists in obtaining  $n$  codified elements from the  $k$  original elements that

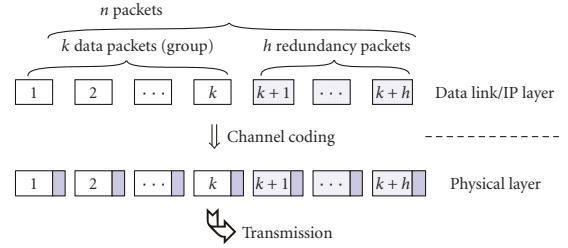


Figure 1. Packet level FEC is applied to groups of packets and bit level FEC is used for channel coding

need to be sent. We define the coding rate as

$$r = \frac{k}{n}. \quad (1)$$

Equivalently, we define the FEC expansion ratio as the inverse of the coding rate. In order to recover the original elements in the receiver (i.e. decode), at least  $k'$  elements must be received, where  $k'$  is equal or greater than  $k$ . We define the protection time as the time needed to send a full block of  $n$  elements.

Obviously, if we need more than  $k$  elements in order to recover the original ones, we will face a coding inefficiency. This inefficiency can be expressed as

$$\eta = \frac{k'}{k}. \quad (2)$$

Inefficiency, as defined in the previous equation, is a number equal or greater than 1. In the optimal case  $\eta = 1$ . Perfectly efficient codes are usually called MDS (Maximum Distance Separable).

Systematic codes (i.e. those where the  $k$  original elements are included among the  $n$  coded elements) are usually preferred, because if not enough elements are received in order to decode, they offer the advantage that among the received elements there will be some of the original elements (which may be useful in real-time applications).

In the previous explanation, we have mentioned elements as the input of the codification process. These elements can be either bits (as normally in channel codification), symbols or packets. Usually, bit level and small block FEC codes are MDS and hence are optimal. But as showed in [2], bit level FEC or small block FEC codes are not powerful enough when we face long signal blockages because the size of the coding block is not usually large enough to offer sufficient protection time. Different levels of coding are illustrated in Figure 1. Although some techniques have been presented in order to use small block codes to encode large amounts of data (see e.g. [2]), they

normally suffer from a lack of flexibility and from the coupon collector problem.

The coupon collector problem (CCP) can be briefly explained in the following way: if we split the data in small subblocks and scramble them before transmission (to avoid an error burst to concentrate in only an encoding subblock), the receiver decoder will have to wait for the last element of the last remaining encoded block while useless extra packets for already decoded subblocks are being received. This is another source of inefficiency, and when the coding block is relatively large (i.e. in the order of 1 MB), CCP inefficiency usually dominates over codification inefficiency [7]. Therefore, the solution is to use large block FEC codes which normally are not MDS but offer arbitrarily large encoding blocks without splitting the data in subblocks and without involving a high computational cost.

## 2.1. LDPC-LDGM Codes

LDPC codes were first introduced by Gallager in 1960 [8], [9] but rested almost forgotten till MacKay and Neal retook them in 1995 [10]. In the last years, they have been the base of Tornado codes, LT codes and Raptor codes.

LDPC codes are systematic linear block codes and are based on a sparse binary parity check matrix. This matrix represents the logic relations between source packets and redundant packets resulting from the encoding. The matrix has dimensions  $h \times n$  where  $h = n - k$  is the number of redundant packets. The  $k$  first columns of the matrix represent the source packets and the last  $h$  columns represent the redundant packets. Each row represents an equation in which packets are bit-wise xored and the result is always a packet whose bits are all 0. If an element in a row is set to 1, it means that the corresponding packet takes part in the equation. There is a dual form of representing the same information using a bipartite graph. There are two types of nodes in these graphs: message nodes (which represent the packets) and check nodes (which represent equations). A line between a message node and a check node is equivalent to a 1 in the matrix.

Back to the matrix representation, the LDPC codification process consists in resolving a linear system, where the unknowns are the redundant packets. It is easy to see that, in a general case, this is a computationally complex task as the matrix has to be inverted and the resulting matrix is not sparse. LDGM matrices are a particular case of LDPC matrices in which the columns corresponding to the redundant packets have been simplified. Three variants of LDGM have been described [5]: plain LDGM, LDGM Stair-

case and LDGM Triangle. In the plain LDGM, the submatrix formed by the columns corresponding to redundant packets is a diagonal matrix. In the case of LDGM Staircase, it is a lower bidiagonal matrix. In LDGM triangle it is a lower bidiagonal matrix with a pseudorandom distribution of 1s in the lower triangle. As the upper triangle of the redundant packets submatrix is empty (all zeroes), the resulting equations are resolvable directly by back-substitution and we can take advantage of the matrix sparsity.

Both a general LDPC and the LDPC-LDGM variants are decodable while the packets are being received. Every received packet is substituted in the equations and it is checked whether any of them can be solved. If a certain number of equations can be solved, a number of original packets smaller than or equal to the number of solved equations may be obtained. In this type of codes, the number of received packets  $k'$  needed to obtain all the original packets is a random number whose distribution depends on the code and the order of arrival of the packets.

## 2.2. Raptor Codes

Raptor codes are a type of fountain codes invented by Shokrollahi [11]. These codes are based in the LT (Luby Transform), which are other fountain codes previously invented by Luby [12]. In fountain codes, the amount of packets resulting from the codification,  $n$ , is not predetermined and the encoder is able to generate any amount of redundant packets. In this kind of codifications, a certain amount of input elements are bit-wise xored to obtain a redundant packet. It is supposed that the decoder will know how each redundant element was generated, which can be done by adding a header to the packets or by time synchronization.

Every time a redundant element is generated, a weight distribution is sampled, returning a weight  $d \in [1, k]$ . Then,  $d$  original elements are pseudo-randomly selected and bit-wise xored to generate the redundant element. The problem of LT codes is that for any amount of generated redundant elements, there is nothing that guarantees that all the original elements have been covered and thus the decoding may be impossible [11]. The risk can be alleviated by increasing the average weight of the redundant elements but this also increases the coding and decoding complexity.

The idea behind Raptor codes is to relax LT codes in such a way that only a certain amount of the input elements are recoverable in the decoding. But, actually we want to recover all the original elements. Raptor codes solve this problem by applying a preliminary LDPC-like codification to the original elements prior to the LT

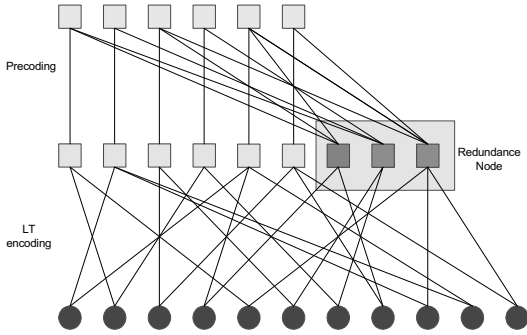


Figure 2. Example of Raptor coding

codification. Thus, the probability of a original element left uncovered is dramatically reduced. Although an additional step is added, the LT can then be simpler and the overall complexity is reduced [11], [13]. A illustration of Raptor coding is showed in Figure 2.

These codes also have another shortcoming: they are generally not systematic. The straight-forward idea of sending the original packets prior to the redundant ones is inefficient because the original packets received would not be useful in the decoding of the missing ones. The solution is to modify the precodification process, the weight distribution and the pseudo-random numbers generator to make the original elements be result of the codification. These changes do not increase the complexity significantly [11].

When working with Raptor codes, there is always the possibility of an unsuccessful decoding process no matter how large the number of received packets. This happens when all the elements are not covered by the received ones, and it is independent of channel-induced errors. Thus, inefficiency must be defined for a certain decoding failure probability, e.g.  $P_f = 10^{-6}$ , which is the probability that some transmitted packets cannot be recovered given a certain number of received packets, without considering channel errors. This does not mean that the packets are going to be lost, but only that some additional received packets will be needed.

### 3. Analysis of LDPC-LDGM vs. Raptor

This section presents the performance of LDPC-LDGM codes and compares it with the known performance of the Raptor code designed for the 3GPP MBMS (Multimedia Broadcast Multicast System) [14]. It should be noted that Raptor codes performance is strongly implementation dependent. The 3GPP MBMS code has been optimized to be efficient even for small amounts of input packets and low computational cost.

Although LDPC-LDGM performace in terms of efficiency does not depend on the size of the input/output packets (nor the Raptor codes one), it is worth saying that all the simulations have been done for 1KB packets. For such a large block FEC codes, the size of packets only affects the coding and decoding times as there will be more xor operations to perform (which grow linearly with the packet size).

### 3.1. LDPC-LDGM performance results

The Inefficiency ratio,  $\eta$ , of LDPC-LDGM codes are basically dependent on the size and the sparsity of the generator matrix [5]. We analyze herein the influence of both parameters. The size is directly determined by the number of input elements,  $k$ , and the FEC expansion ratio,  $R = \frac{1}{r}$ . The sparsity is a design parameter of the code, but results in [5] show that 3 ones per column is the best trade-off between complexity and performance. The dependence of LDPC-LDGM codes performance on the amount of input packets and the FEC expansion ratio has been simulated using 1000 Monte Carlo runs.

The inefficiency of LDPC-LDGM codes as a function of the FEC expansion ratio when  $k = 10000$  is shown in Figure 3. For all the three codes inefficiency grows linearly with the FEC expansion ratio. But while the results of Staircase and Triangle codes are similar, results of the plain LDGM are much worse and inefficiency grows faster. With a less sparse matrix, better results can be obtained for the plain LDGM, but the increase in the complexity would make the code unusable [5], and even then Triangle and Staircase codes would still be better.

In the considered scenario, LDGM Triangle slightly outperforms LDGM Staircase. This comes at the cost of having a less sparse matrix in the columns corresponding to redundant packets.

The dependence of the inefficiency with the amount of input packets when the FEC expansion ratio is 1.5 is shown in Figure 4, together with the 99% confidence interval. In this case, LDGM Triangle and LDGM Staircase also outperform the plain LDGM. While Staircase and Triangle codes improve their inefficiency as  $k$  grows, the plain LDGM inefficiency gets even worse. LDGM Triangle and LDGM Staircase inefficiencies tend respectively to the values of 1.055 and 1.068.

The conclusion drawn from these results is that LDGM is not competitive with respect to the Staircase and Triangle versions. Therefore, the plain LDGM will not be taken into account in the comparison

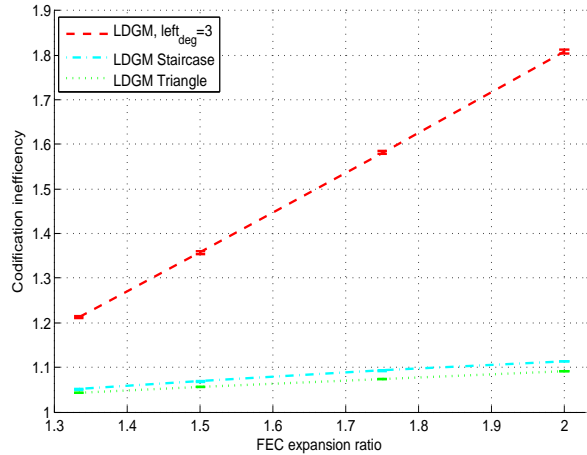


Figure 3. Coding inefficiency of LDGM, LDGM Staircase and LDGM Triangle as a function of FEC expansion ratio, when  $k = 10000$  (99% confidence intervals are included)

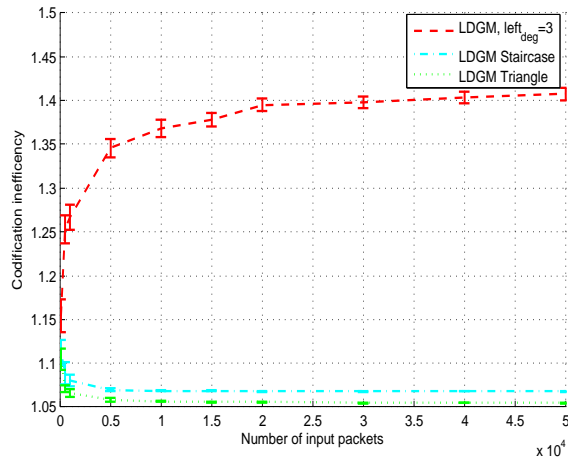


Figure 4. Coding inefficiency of LDGM, LDGM Staircase and LDGM Triangle as a function of  $k$ , when FEC expansion ratio is 1.5 (99% confidence intervals are included)

with Raptor codes. Another remarkable fact is that although LDGM Triangle results are better than LDGM Staircase ones, the encoding/decoding complexity of LDGM Triangle is higher since its matrix is less sparse (i.e. the lower triangle of the redundant packets submatrix is randomly filled).

### 3.2. 3GPP MBMS Raptor performance

In the literature, many papers have dealt with Raptor codes performance, see e.g. [13] and references therein. In this document, we are going to focus on

the performance of the code designed for the 3GPP MBMS, which is a very optimized one and similar to the one potentially needed by a mobile satellite communications system. In [14], an in-depth analysis of this code is done and a formula that closely approximates the decoding failure probability for  $k > 200$  is presented:

$$P_f = \begin{cases} 1 & \text{if } m < k \\ 0.85 \cdot 0.567^{m-k} & \text{if } m \geq k \end{cases}, \quad (3)$$

where  $m$  is the number of received packets. From this formula we can easily obtain an expression of inefficiency as a function of the number of input elements, for a given failure probability  $P_f$ :

$$\eta = \frac{\log(P_f/0.85)}{k \log 0.567} + 1, \text{ if } m \geq k. \quad (4)$$

On the other hand, Raptor codes do not have any dependence on the FEC expansion ratio, as a Raptor encoder can generate any number of redundant packets.

### 3.3. Performance Comparison

The inefficiency as a function of the number of input packets for three large block FEC codes: LDGM Staircase, LDGM Triangle and the 3GPP MBMS Raptor is shown in Figure 5. We can see how the Raptor code clearly outperforms LDGM Staircase and Triangle. For example, when  $k = 10000$  (block size 10MB) we see that Raptor inefficiency is 1.002 while LDGM Triangle is 1.055 and LDGM Staircase is 1.068. It is also important to note that inefficiency of Raptor codes tends to be almost 1 as the number of input elements grows. Moreover, as it can be seen in Figure 6, while Raptor code inefficiency does not depend on the FEC expansion ratio, LDGM Staircase and Triangle inefficiencies get even worse if we use higher expansion ratios.

## 4. Conclusions

We conclude that Raptor codes clearly outperforms any variant of LDPC-LDGM codes and in almost every aspect. Firstly, Raptor codes inefficiency is lower than that of the LDPC-LDGM codes regardless of the number of input packets and secondly, the performance difference widens when a high FEC expansion ratio is needed. Considering this particular aspect, Raptor codes would be the best option for instance for satellite mobile communications.

In favour of LDPC LDGM one can say mention that they are very efficient when the FEC expansion

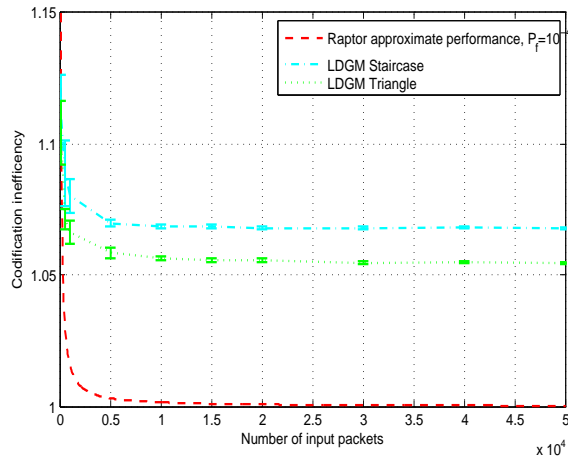


Figure 5. Codification inefficiency of LDGM Staircase, LDGM Triangle and 3GPP MBMS Raptor as a function of  $k$ , when FEC expansion ratio is 1.5 for the LDGM codes (99% confidence intervals are included)

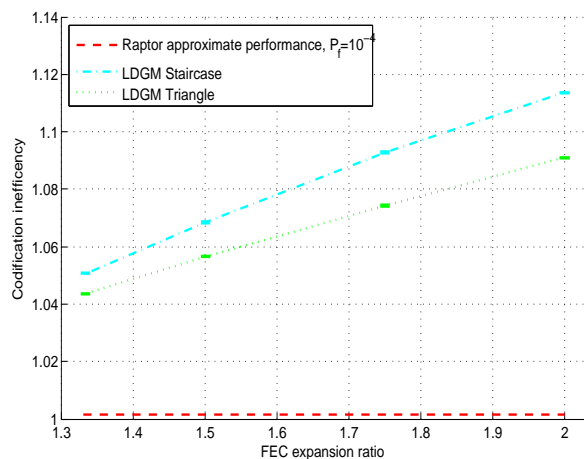


Figure 6. Codification inefficiency of LDGM Staircase, LDGM Triangle and 3GPP MBMS Raptor as a function of FEC expansion ratio, when  $k = 10000$  (99% confidence intervals are included)

ratio is low (i.e.  $\eta < 1.07$  when  $R = 1.5$  and  $k = 10000$ ). Moreover, LDPC-LDGM implementation is freely available.

## References

[1] “Digital video broadcasting - handheld (dvb-h),” ETSI standard EN 302 304, 2004.  
 [2] S. Cioni, C. P. Niebla, G. Seco-Granados, S. Scalise, A. Vanelli-Coralli, and M. Vázquez-Castro, “Advanced Fade Countermeasures for DVB-S2 Systems in Railway

Scenarios,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, pp. 1–17, 2007.

- [3] S. Cioni, A. Vanelli-Coralli, G. Corazza, J. López-Vicario, and M. Vázquez-Castro, “SIMO Diversity with Antenna Selection for DVB-S2/RCS in LMS Scenarios,” *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pp. 255–258, Sept. 2007.  
 [4] J. L. Vicario, M. Vázquez-Castro, G. Seco-Granados, S. Cioni, A. Vanelli-Coralli, and G. Corazza, “Hybrid Selection/MRC for DVB-S2/RCS in Land Mobile Satellite Scenarios,” in *AIAA International Communications Satellite Systems Conference (ICSSC)*, 2007, pp. 1–9.  
 [5] V. Roca and C. Neumann, “Design, Evaluation and Comparison of Four Large Block FEC Codes, LDPC, LDGM, LDGM Staircase and LDGM Triangle, plus a Reed-Solomon Small Block FEC Codec,” INRIA, Tech. Rep., 2004. [Online]. Available: [citeseer.ist.psu.edu/roca04design.html](http://citeseer.ist.psu.edu/roca04design.html)  
 [6] U. Demir and O. Aktas, “Raptor versus Reed Solomon Forward Error Correction Codes,” *2006 International Symposium on Computer Networks*, pp. 264–269, 2006.  
 [7] R. Firrincieli, G. Albertazzi, G. Corazza, and V. Coralli, “Packet Layer FEC for SDBM Services,” in *AIAA International Communications Satellite Systems Conference (ICSSC)*, 2005.  
 [8] R. Gallager, “Low-Density Parity-Check Codes,” Ph.D. dissertation, Massachusetts Institute of Technology, 1960.  
 [9] —, “Low-Density Parity-Check Codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.  
 [10] D. MacKay and R. Neal, “Good Codes based on Very Sparse Matrices,” *Cryptography and Coding: 5th IMA Conference*, 1995.  
 [11] A. Shokrollahi, “Raptor Codes,” *IEEE/ACM Transactions on Networking (TON)*, vol. 14, pp. 2551–2567, 2006.  
 [12] M. Luby, “LT Codes,” *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pp. 271–280, 2002.  
 [13] P. Cataldi, M. Shatarski, M. Grangetto, and E. Magli, “Implementation and Performance Evaluation of LT and Raptor Codes for Multimedia Applications,” *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2006, pp. 263–266, 2006.  
 [14] M. Luby, M. Watson, T. Gasiba, and T. Stockhammer, “Mobile Data Broadcasting over MBMS Tradeoffs in Forward Error Correction,” *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, 2006.