

# Impacts of the Packet Scheduling on the Performance of Erasure Codes: Methodology and Application to GLDPC-Staircase Codes

Ferdaouss Mattoussi, Vincent Roca, Bessem Sayadi

## ► To cite this version:

Ferdaouss Mattoussi, Vincent Roca, Bessem Sayadi. Impacts of the Packet Scheduling on the Performance of Erasure Codes: Methodology and Application to GLDPC-Staircase Codes. IEEE. IEEE European Conference on Networks and Communications (EUCNC'15), Jun 2015, Paris, France. 2015, <<http://eucnc.eu>>. <hal-01144380>

HAL Id: hal-01144380

<https://hal.inria.fr/hal-01144380>

Submitted on 21 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Impacts of the Packet Scheduling on the Performance of Erasure Codes: Methodology and Application to GLDPC-Staircase Codes

Ferdaouss Mattoussi  
Institut Fresnel, GSM research Team  
Marseilles, France  
Email: ferdaouss.mattoussi@fresnel.fr

Vincent Roca  
Inria, PRIVATICS research Team  
Grenoble, France  
Email: vincent.roca@inria.fr

Bessem Sayadi  
Alcatel-Lucent Bell labs  
Paris, France  
Email: bessem.sayadi@alcatel-lucent.com

**Abstract**—Content broadcasting over wireless networks heavily relies on Application-Level FEC codes to improve transmission robustness in front of channel erasures. Because they operate in the higher layers of the protocol stack, they benefit from a lot of flexibility. In particular, since streaming applications and bulk transfer applications have different constraints, different packet scheduling strategies may be used by the sender, offering different trade-offs between decoding latency and long erasure burst protection. This work tries to find the best packet scheduling scheme(s) at a sender for a given type of AL-FEC codes.

The contributions are twofold: first we define a methodology to measure the impacts of packet scheduling on AL-FEC performance, both under Iterative (IT) and Maximum Likelihood (ML) decoding, for a large set of channels; then we apply this methodology to GLDPC-Staircase codes, an extension of LDPC-Staircase codes using Reed Solomon codes as inner codes. In previous works we showed that these codes have erasure recovery performance close to ideal codes when packets are transmitted in a random order. In this work we show that these codes perform extremely well when sending source packets sequentially first (a key requirement to keep latency minimum with streaming applications) and then extra-repair packets followed by LDPC repair packets, both in a random order.<sup>1</sup>

## I. INTRODUCTION

### A. About LDPC and Generalized LDPC-Staircase codes

Low Density Parity Check (LDPC) codes have been intensively studied due to their near-Shannon limit performance under iterative (or Belief-Propagation) decoding [1], [2]. An  $(N, K)$  LDPC code, where  $N$  is the code length and  $K$  is its dimension, can be graphically represented as a bipartite graph with  $N$  “variable nodes” (VNs) and  $M = N - K$  “check nodes” (CNs). A CN of degree  $n$  can be interpreted as a Single Parity Check (SPC) code  $(n, n - 1)$ , i.e. as a linear block code associated with one parity equation.

In order to improve error floor and minimal distance performances of LDPC codes, a generalization of these codes was suggested by Tanner in [2], so-called Generalized LDPC (or GLDPC) codes, for which subsets of the variable nodes obey a more complex constraint than an SPC constraint. More precisely, the SPC check nodes are replaced by a generic linear block codes  $(n, k)$  referred to as sub-codes or component

codes. Various GLDPC code, for several types of channels, have been designed. The constructions differ either by the components codes (e.g., [3], [4], [5], [6], [7], [6], [8], [9], [10]), and/or by the distribution of the structure of GLDPC codes ([5]), in order to offer a good trade-off between waterfall performance and error floor under an iterative decoding.

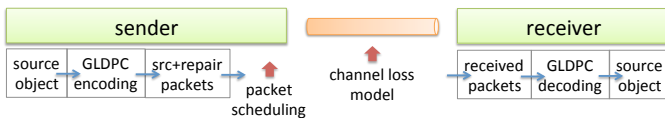
In [11] we proposed a construction for Generalized LDPC codes, using LDPC-Staircase [12] as base code and Reed-Solomon (RS) as component codes. This construction allows each RS component code to produce a potentially large number of repair symbols, called *extra-repair symbols*, on demand. This is well suited to situations where the channel conditions can be worse than expected, or to fountain like content distribution applications, since very small code rates are easily achievable. GLDPC-Staircase code performance is assessed through different types of decoders: via Iterative (IT)+RS decoding (a sub-optimal but fast approach), or Maximum Likelihood (ML) decoding (fast but also costly), or hybrid-decoding which uses both techniques in sequence if IT+RS decoding is not sufficient (see [11] for more information on IT+RS/ML decoding). Simulation results and theoretical analyses show that this scheme features excellent decoding performance, i.e. good waterfall region and small error floors, and they approach channel capacity even for very small objects.

### B. Target use-cases: digital content broadcast services

Such a content delivery system as the 3GPP “Multimedia Broadcast/Multicast Service” (MBMS) [13] is characterized by the fact that there is no back channel, and therefore it offers an unlimited scalability in terms of number of receivers who behave in a completely asynchronous way. This system sees the transmission channel as a “packet erasure channel”, with packets that either arrive (without any error) to the destination or are lost (for instance because of router congestion problems or severe wireless reception conditions).

Since there is no backward channel, no repeat request mechanism can be used that would enable the source to adapt its transmission according to the feedbacks sent by receivers. Therefore these systems rely on the Asynchronous Layered Coding (ALC) [14] reliable multicast transmission protocol along with the FLUTE file delivery application [15]. To be robust in front of erasures, FEC encoding is performed within ALC and repair packets are transmitted along with

<sup>1</sup>This work was supported by the ANR-09-VERS-019-02 grant (ARSSO project) and by the Inria - Alcatel Lucent Bell Labs France joint laboratory.



**Fig. 1:** The various steps considered in the transmission chain.

the original data packets. Thanks to this redundancy, up to a certain number of missing packets can be recovered at the receivers, and a given repair packet can recover different erased packets at different receivers. The GLDPC-Staircase codes considered in this work are such a potential FEC solution (known as Application-Level FEC, or AL-FEC) for the MBMS services, like other already standardized and used codes (e.g., Raptor(Q), LDPC-Staircase, Reed-Solomon) [16].

### C. Problem position

Figure 1 illustrates the situation: the sender encodes the source object (e.g., a file with FLUTE, or a continuous video flow with a streaming application) and chooses a packet scheduling. This packet flow is subject to erasures while traveling to a given receiver. This latter performs FEC decoding and (hopefully) recovers the original object. So, for a given type of AL-FEC codes, two key questions are:

- how is the performance impacted by the packet transmission schedule, i.e. the order in which packets are sent over the channel?
- how is the performance impacted by the packet loss distribution observed by the receivers (e.g. a memory-less channel with erasures uniformly spread, versus a channel with erasure bursts of a certain length)?

As we consider broadcast/multicast services, different receivers usually observe different channels (e.g., the short/long term fading properties of a moving receiver can change significantly, unlike that of a fixed receiver in line-of-sight conditions). Since there are as many channels as there are receivers, *any AL-FEC performance analysis must consider a large set of channels.*

Certain AL-FEC codes do not depend on the packet scheduling. This is the case of ideal (i.e., Maximum Distance Separable) codes (e.g., Reed-Solomon) that perform identically in terms of erasure recovery performance no matter the subset of packets received, as long as a minimum number of  $k$  (code dimension) packets among  $n$  (code length) are available. This is also the case of codes that by construction do not show any dependency over the set of packets received, because each encoding packet is produced in the same manner from a set of intermediary symbols: Raptor [17] and RaptorQ [18] have this property. Finally non systematic AL-FEC codes can feature a low dependency on packet scheduling since by definition only repair packets are available at a receiver. On the opposite, certain codes create strong relationships between repair packets: e.g., with LDPC-Staircase codes (more generally repeat-accumulate codes), each repair packet is the sum of the previous repair packet with some additional source packets. In that case the subset of packets received does potentially impact the code performance.

Another dimension to the problem is the type of AL-FEC decoding performed: on the fly Iterative (IT) decoding, as

packets arrive, does not behave the same as maximum Likelihood (ML) decoding. In practice, there is a clear incentive in using IT decoding whenever possible because of its low complexity. ML decoding is on the opposite CPU intensive, even if techniques like Structured Gaussian Elimination (SGE) significantly reduce the gap between IT and ML[16]. In this work we consider both decoders since a practical codec will probably start with IT decoding and finish with ML decoding (using SGE) if needed. Therefore *a good packet scheduling for a given AL-FEC code must perform well both under IT and ML decoding.*

### D. Contributions of the work

The contributions of this work are twofold. First of all, we define a methodology to measure the impacts of packet scheduling on any AL-FEC code both under Iterative (IT) and/or Maximum Likelihood (ML) decoding, for a large variety of bursty erasure channels characterized by a finite-state Markov chain. This methodology will be detailed in Section II.

We then apply this methodology to the GLDPC-Staircase AL-FEC codes. In our previous works, we have shown that under ML decoding, these codes have erasure correction capabilities close to that of ideal codes *when packets are transmitted in a random order*, no matter the object size. We now show that GLDPC-Staircase codes can keep an exceptional performance level both under IT and ML decoding when sending source packets sequentially first (a key requirement with streaming applications to keep the decoding delay as small as possible), and then extra-repair packets followed by repair packets, both in random order. This scheduling offers the best compromise between low decoding failure probability and low impact of the erasure burst length. This will be discussed in Section III.

## II. PROPOSED PERFORMANCE EVALUATION METHODOLOGY

### A. Considering many different channels with the Gilbert model

Section I-C has stressed the importance of testing a given packet scheduling across a very large set of channels. To that goal we use the two-state Markov model known as Gilbert model[19], a choice in line with many works that showed this model to be a good approximation to capture the temporal dependencies of packet losses [20][21][22].

This model is either characterized by the transition probabilities between the two states,  $p$  and  $q$ , or by the mean Packet Loss Ratio,  $PLR$  and Average Burst Length,  $ABL$ , with [23]:

$$PLR = \frac{p}{p+q}, \quad ABL = \frac{1}{q}. \quad (1)$$

When  $q = 1-p$ , then it is equivalent to a memory-less channel.

### B. Evaluation methodology

To consider a large number of channels, we vary the  $p, q$  parameters as follows. For each transmission scheduling scheme, we consider 21 values between 0 and 1 for both  $p$  and  $q$  (i.e., every 0.05)<sup>2</sup>. It results in a  $21 \times 21$  grid, or 441 different channels. Since simulations follow a stochastic approach, for

<sup>2</sup>Of course a smaller granularity can be chosen for more detailed results.

each channels (i.e.,  $p, q$  pair), we perform 1000 tests, for a total of 441000 tests. Some of these tests are successful and enable a receiver to decode and recover the original object, while others fail. *The best packet scheduling for a given AL-FEC code is therefore the one that features the highest total number of successful tests over all the channels considered.*

As explained above, the Gilbert model is also totally determined by the  $PLR, ABL$  tuple. Therefore it is possible to carry out the same tests by varying the  $PLR$  and  $ABL$  parameters between some predefined bounds and to count the number of successful decodings for each of the 1000 tests carried out for a given channel. The results are equivalent, but the presentation is perhaps easier to interpret as the  $PLR, ABL$  parameters are more intuitive than the  $p, q$  parameters. In our case we found easier to perform all the simulations through the  $p, q$  parameters, storing the results in an SQL database, and doing the appropriate queries to compute the  $PLR, ABL$  results (see Sections III-A and III-B).

*a) About impossible decoding areas in the  $(p, q)$  grid:*

There are areas in the  $21 \times 21$  grid where decoding is in theory not feasible as the number of packets received ( $n * (1 - PLR)$ ) is lower than  $k$ . However, since we are doing stochastic simulations using  $p, q$  as transition probabilities, the actual number of packets received is not necessarily exactly equal to  $n * (1 - PLR)$  and decoding may still be feasible.

*b) About overhead based evaluation methods:*

Our erasure recovery performance evaluation is not based on overhead evaluation techniques, i.e., on the determination of the minimum number of packets that need to be received in excess to  $k$  for decoding to succeed. This is deliberate<sup>3</sup>.

### C. Transmission scheduling schemes considered

In order to analyze the performance specific AL-FEC codes, several transmission schedulings must be identified. In our case, with GLDPC-Staircase codes, we identified the following 22 different schedulings, classified into four categories:

- Category 1 where all the packets sent randomly. This is the reference scheduling that we used in [11];
- Category 2 where source packets are sent first, *sequentially*, followed by repair packets using different orderings. As it minimizes delays (there is no need to buffer all the source packets before beginning the transmissions), it is well suited to streaming;
- Category 3 where source packets are sent first *randomly* followed by repair packets using various orderings;
- Category 4 where source and repair packets are sent randomly in blocks (i.e., source, LDPC repair and extra-repair packets);

Table I gives more details. Although the list is not exhaustive, we believe it covers a large set of possibilities, in particular those of benefit to file download and streaming applications.

<sup>3</sup>E.g., we identified a scheduling with excellent overheads for low to medium  $PLR$ , but behaving badly when approaching the theoretical limit. When considering the total number of successful tests, this "promising" solution turned out to be worse than alternative scheduling.

### D. Complementarity with other evaluation methods

This methodology is complementary with other techniques:

- the Density Evaluation (DE) and EXIT performance evaluation methods enable an **asymptotic analysis**, when the code length tends to infinity and assuming there is no cycle in the graph. There is no channel loss nor packet transmission model, the loss rate being the only parameter of interest. These techniques are excellent to assess the theoretical performance of a code family, and we already applied both methods to GLDPC-Staircase codes [24][25];
- the **finite size performance evaluation methods** enable a detailed performance analysis, and such metrics as the average decoding overhead or the decoding failure probability are evaluated through a stochastic approach. In [11] we analyze GLDPC-Staircase performance assuming a fully random packet scheduling for a wide range of parameters like the code dimension, code rate, or various internal parameters. Because the packet scheduling is random, the results only depend on the average packet loss rate;

The methodology proposed here nicely fits in the panorama: it is a finite size evaluation that assesses the code performance with different channel loss models and packet scheduling approaches, which is ignored in the above techniques.

## III. APPLICATION TO GLDPC-STAIRCASE CODES

Let us consider an object composed of  $k = 1,000$  packets. Let us consider GLDPC-Staircase codes, with a base code rate (i.e. associated to the LDPC-Staircase code) equal to  $r_L = \frac{2}{3}$  and a global code rate either equal to  $r_G = \frac{1}{2}$  or  $r_G = \frac{1}{3}$ . Therefore, with  $r_G = \frac{1}{2}$ , we have 500 LDPC repair packets (i.e. produced by LDPC-Staircase encoding) and 500 extra-repair packets (i.e. produced by Reed-Solomon encoding). With  $r_G = \frac{1}{3}$ , we still have 500 LDPC repair packets but we now have 3 times more extra-repair packets, namely 1500 extra-repair packets. In the following, we first provide general results obtained while varying the  $(p, q)$  parameters, then complementary results while varying the  $(PLR, ABL)$  parameters. If the two methods are equivalent in theory, they can highlight behaviours that would not be visible otherwise.

### A. Results with respect to the $(p, q)$ parameters

Tables II and III provide the total number of successful decodings (among a total of 441,000 tests), for which all the erased source packets can be recovered, as a function of the transmission scheduling, for code rates  $r_G = \frac{1}{2}$  and  $r_G = \frac{1}{3}$  and for IT+RS and ML decoding techniques. Of course, the higher the number in the table, the better, in the sense that this packet scheduling enables FEC decoding to take place successfully for a higher number of different channels.

1) *The case of IT+RS decoding:* From table II we see that transmitting the LDPC repair packets just after the data packets (Tx-modes 1, 4, 5, 8, 11, 13, 14 and 15) leads to very poor results with IT+RS decoding. On the opposite, sending the extra-repair packets (randomly or sequentially) just after the data packets (Tx-modes 2, 3, 6, 7, 9, 10, 12, 16, 17 and

|            |            |   |
|------------|------------|---|
| Category 1 |            | <b>all packets are sent randomly. This is the reference scheduling</b>  |
|            | Tx-Mode-0  | Randomly send all source and repair packets (fully random order)  |
| Category 2 |            | <b>source packets are sent first sequentially, followed by repair packets in a certain order</b>              |
|            | Tx-Mode-1  | LDPC repair first sequentially and then extra-repair packets sequentially                                     |
|            | Tx-Mode-2  | Fully random order of all repair packets  |
|            | Tx-Mode-3  | Extra-repair packets first sequentially and then LDPC repair packets sequentially                             |
|            | Tx-Mode-4  | LDPC repair packets first randomly and then extra-repair packets sequentially                                 |
|            | Tx-Mode-5  | LDPC repair packets first sequentially and then extra-repair packets randomly                                 |
|            | Tx-Mode-6  | Extra-repair packets first randomly and then LDPC repair packets sequentially                                 |
|            | Tx-Mode-7  | Extra-repair packets first sequentially and then LDPC repair packets randomly                                 |
|            | Tx-Mode-8  | LDPC repair packets first randomly and then extra-repair packets randomly                                     |
|            | Tx-Mode-9  | Extra-repair packets first randomly and then LDPC repair packets randomly                                     |
| Category 3 |            | <b>source packets are sent first randomly, followed by repair packets in a certain order</b>                  |
|            | Tx-Mode-10 | Fully random order of all repair packets  |
|            | Tx-Mode-11 | LDPC repair packets first randomly and then extra-repair packets randomly                                     |
|            | Tx-Mode-12 | Extra-repair packets first randomly and then LDPC repair packets randomly                                     |
|            | Tx-Mode-13 | LDPC repair packets first sequentially and then extra-repair packets sequentially                             |
|            | Tx-Mode-14 | LDPC repair packets first randomly and then extra-repair packets sequentially                                 |
|            | Tx-Mode-15 | LDPC repair packets first sequentially and then extra-repair packets randomly                                 |
|            | Tx-Mode-16 | Extra-repair packets first sequentially and then LDPC repair packets sequentially                             |
|            | Tx-Mode-17 | Extra-repair packets first randomly and then LDPC repair packets sequentially                                 |
|            | Tx-Mode-18 | Extra-repair packets first sequentially and then LDPC repair packets randomly                                 |
| Category 4 |            | <b>source and repair packets are sent randomly per block type (i.e. source, LDPC repair, extra-repair)</b>    |
|            | Tx-Mode-19 | Randomly send extra-repair packets first, then randomly LDPC repair packets, and then randomly source packets |
|            | Tx-Mode-20 | Randomly send LDPC repair packets first, then randomly extra-repair packets, and then randomly source packets |
|            | Tx-Mode-21 | Randomly send all GLDPC repair packets first, then randomly source packets                                    |

**TABLE I:** The 22 packet schedulings considered in this work.

| Category (2)           | 1       | 2       | 3       | 4       | 5      | 6       | 7       | 8       | 9              |
|------------------------|---------|---------|---------|---------|--------|---------|---------|---------|----------------|
| $r_G=1/3$              | 87,673  | 224,636 | 231,169 | 86,933  | 87,671 | 238,336 | 235,796 | 86,952  | <b>241,383</b> |
| $r_G=1/2$              | 80,140  | 135,920 | 142,678 | 86,837  | 87,642 | 142,997 | 143,654 | 86,854  | <b>147,726</b> |
| Category (3)           | 10      | 11      | 12      | 13      | 14     | 15      | 16      | 17      | 18             |
| $r_G=1/3$              | 224,492 | 86,892  | 241,269 | 87,245  | 86,652 | 87,678  | 231,214 | 238,322 | 235,804        |
| $r_G=1/2$              | 136,048 | 86,869  | 143,795 | 87,190  | 86,653 | 87,359  | 142,376 | 143,002 | 143,614        |
| Categories (1) and (4) | 0       | 19      | 20      | 21      |        |         |         |         |                |
| $r_G=1/3$              | 237,238 | 241,875 | 241,851 | 241,966 |        |         |         |         |                |
| $r_G=1/2$              | 142,037 | 143,883 | 143,816 | 143,858 |        |         |         |         |                |

**TABLE II:** Total number of successful decodings for the various schedulings under IT+RS decoding.

18) is much more efficient. We also note that sending extra-repair or LDPC repair in a random order has a positive impact. Therefore it appears that Tx-mode=9 and 12 are the best, with the additional benefit for Tx-mode=9 that source packets are sent in sequence first, which avoids extra delays at the sender.

Let us now consider the cases where data packets are not sent at the beginning, i.e. the scheduling categories (1) and (4). Table II proves that sending the repair packets (totally randomly or randomly per block type) performs extremely well. However these schedulings require that the sender knows all the source packets first before being able to perform FEC encoding, and therefore transmitting packets. This is an issue in terms of delay if source packets arrive progressively, as is the case of streaming applications.

*Therefore, in order to obtain good performance under (IT+RS) decoding while keeping delays to a minimum, we recommend using Tx-mode=9.*

2) *The case of ML decoding:* Table III shows that under ML decoding the performance differences between the various scheduling modes is less significant than for IT+RS decoding (the extreme are for Tx-mode=1 and Tx-mode=13, i.e., a 3356 difference or 1.04%). However it confirms that with ML decoding too, the Tx-mode=9 approach performs extremely well (it is only 0.09% behind the optimum).

*All things considered, it appears that Tx-mode=9 is the*

*packet scheduling that offers the best compromise for both IT+RS and ML decoding.*

### B. Results with respect to the (PLR, ABL) parameters

Let us now consider the *PLR, ABL* parameters. In order to assess the impacts of packet loss bursts on performance, we considered  $ABL = \{1, 1.5, 2, \dots, 20\}$  at  $PLR = \{33.33\%, 50\%, 66.66\%\}$  for both IT+RS decoding and ML decoding<sup>4</sup>.

For IT+RS decoding only, decoding becomes impossible at  $PLR = 66.66\%$  (not shown). At  $PLR = 50\%$ , Figures 2 highlight major performance differences. We see that Tx-mode=9 performs reasonably well. Even if it is far from being the most efficient scheduling when the *ABL* increases where Tx-mode=3 and 7 offer a better protection, Tx-mode=9 behaves better with very small values of *ABL*. We also see that categories (1) and (4) perform well, as was the case in Table II. In any case one needs to be very careful because the present results represent only a fraction of the results of Section III-A that consider all the 441 channels.

Let us now consider ML decoding. Our analysis (not shown here) shows that all the packet schedulings behave exactly the

<sup>4</sup>As explained before, tests are conducted only once for the 441 channels. Then we extract all the tests for which  $PLR = p/(p+q) = 2/3$  exactly. Because of the way we iterate over  $(p, q)$ , many such cases exist in the database. This is not necessarily the case for other values of *PLR* and running new tests could be required.

| Category (2)           | 1       | 2       | 3       | 4              | 5       | 6       | 7       | 8       | 9              |
|------------------------|---------|---------|---------|----------------|---------|---------|---------|---------|----------------|
| $r_G=1/3$              | 323,938 | 325,923 | 325,893 | 325,936        | 325,927 | 325,924 | 325,945 | 325,951 | <b>325,974</b> |
| $r_G=1/2$              | 200,487 | 220,769 | 221,245 | 220,643        | 221,118 | 221,112 | 220,647 | 221,766 | <b>220,773</b> |
| Category (3)           | 10      | 11      | 12      | 13             | 14      | 15      | 16      | 17      | 18             |
| $r_G=1/3$              | 325,923 | 325,918 | 325,919 | <b>327,294</b> | 326,283 | 326,014 | 326,088 | 326,029 | 326,089        |
| $r_G=1/2$              | 220,757 | 220,734 | 220,766 | <b>221,096</b> | 220,745 | 221,232 | 219,552 | 221,315 | 220,802        |
| Categories (0) and (4) | 0       | 19      | 20      | 21             |         |         |         |         |                |
| $r_G=1/3$              | 325,905 | 325,918 | 325,919 | 325,917        |         |         |         |         |                |
| $r_G=1/2$              | 220,788 | 220,773 | 220,737 | 220,725        |         |         |         |         |                |

**TABLE III:** Total number of successful decodings for the various schedulings under ML decoding.

same at  $PLR = 33\%$  and  $50\%$ , and Figures 2 show they behave similarly with  $PLR = 66.66\%$ . This is perfectly in line with the results of Section III-A. Note that we are very close to the Shannon limit when  $r_G = \frac{1}{3}$ . The fact that decoding is still feasible with probability 0.5 (approximately) up to  $ABL = 20$  packets is an excellent result that proves the high potential of GLDPC-Staircase codes.

#### IV. CONCLUSIONS

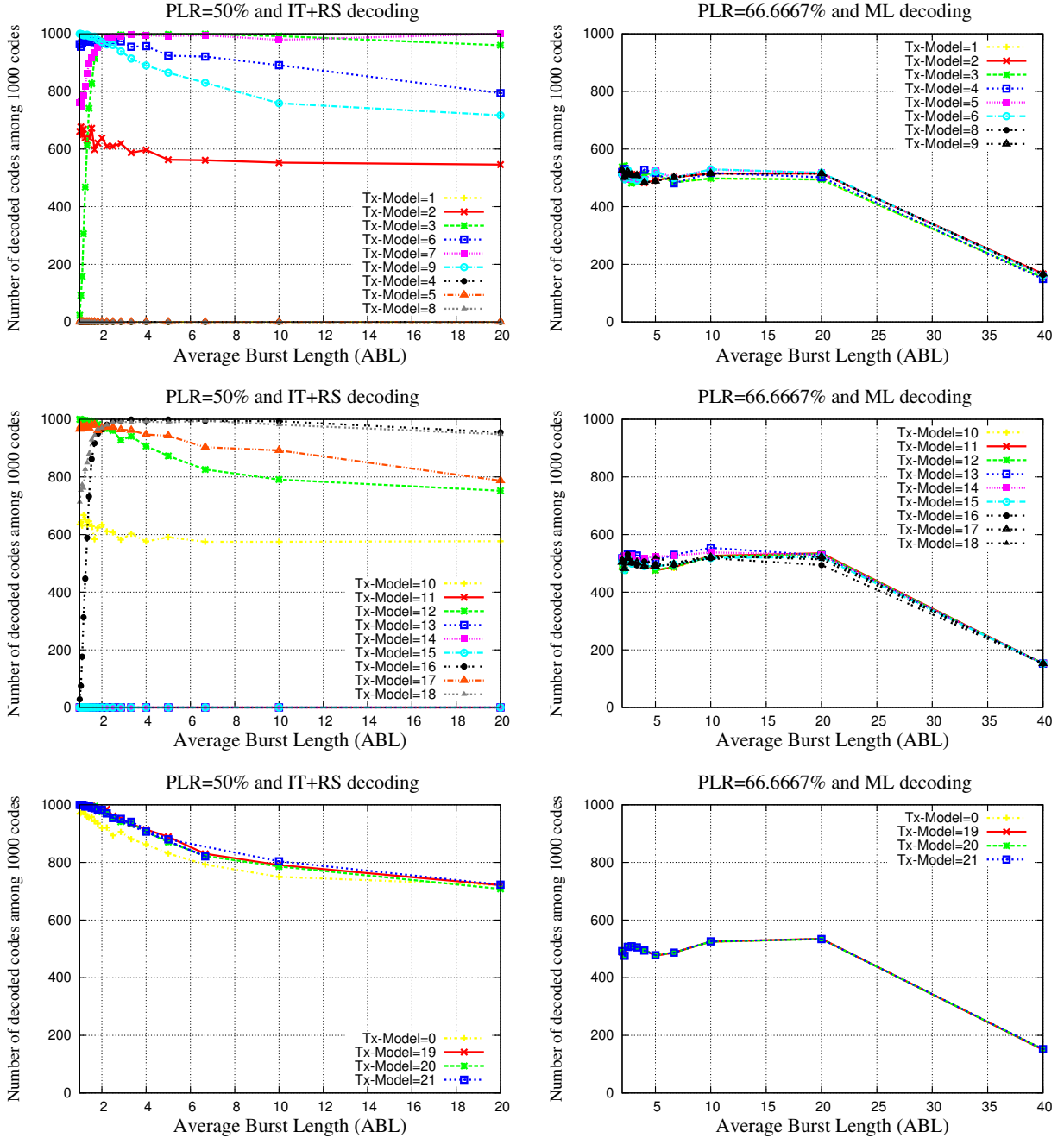
This work investigates the performance of AL-FEC codes as a function of the order under which source and repair packets are transmitted, in the context of multicast/broadcast reliable content delivery protocols. Since different users experience different reception conditions and erasure patterns, we consider a very broad range of channels (i.e.,  $21 \times 21 = 441$  channels) using the two state Gilbert model. We show that this analysis is essential to optimally use certain AL-FEC codes for which the identity of the received packets does matter. This analysis is complementary to the theoretical asymptotic analyses (Density Evaluation and EXIT techniques) and finite size stochastic methods where a memory-less channel only characterized by the packet loss rate is assumed.

The contributions of this work are twofold: (1) we detail a new methodology to address the problem, and (2) we apply this methodology to GLDPC-Staircase codes. By comparing the total number of successful decodings over all the channels considered, we show that a scheduling where source packets are sent first, in sequence, followed by extra-repair packets, randomly, and finally by LDPC repair packets, randomly, leads to excellent performance both under IT+RS and ML decoding. It also has the key advantage of keeping the end-to-end delay as small as possible since transmissions take place as soon as source packets are available (no need to wait for all the source packets to be available for encoding to take place). This is a key advantage with real-time streaming applications.

To the best of our knowledge it is the first time this question is addressed. Note that this is a new question specific to AL-FEC codes, since their flexibility enables a sender to select the best packet scheduling scheme according to the application requirements and AL-FEC features. No such flexibility exists with physical layer FEC codes!

#### REFERENCES

- [1] D. MacKay and R. Neal, "Near Shannon limit performance of low density parity check codes," *IET Electronics Letters*, vol. 33, no. 6, pp. 457–458, 1997.
- [2] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
- [3] J. Boutros, O. Pothier, and G. Zémor, "Generalized Low Density (Tanner) Codes," *IEEE Int. Conference on Communications (ICC'99)*, 1999.
- [4] M. Lentmaier and K. Zigangirov, "On Generalized Low-Density Parity-Check Codes Based on Hamming Component Codes," *IEEE Transactions on Communications*, vol. 3, no. 8, pp. 248–250, 1999.
- [5] G. Yue, L. Ping, and X. Wang, "Generalized Low-Density Parity-Check Codes Based on Hadamard Constraints," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 1058–1079, 2007.
- [6] J. Chen and R. Tanner, "A Hybrid Coding Scheme for the Gilbert Elliott Channel," *IEEE Transactions on Communications*, vol. 54, no. 10, pp. 1787–1796, 2006.
- [7] N. Miladinovic and M. Fossorier, "Generalized LDPC Codes with Reed-Solomon and BCH Codes as Component Codes for Binary Channels," *IEEE Global Telecommunications Conference (GLOBECOM'05)*, Dec. 2005.
- [8] I. Djordjevic, O. Milenkovic, and B. Vasic, "Generalized Low-Density Parity-Check Codes for Optical Communication Systems," *Lightwave Technology*, vol. 23, no. 5, pp. 1939–1946, 2005.
- [9] Y. Wang and M. Fossorier, "Doubly Generalized LDPC Codes," *IEEE Int. Symposium on Information Theory*, pp. 669–673, 2006.
- [10] E. Paolini, M. Fossorier, and M. Chiani, "Analysis of Doubly-Generalized LDPC Codes with Random Component Codes for the Binary Erasure Channel," *Proceedings of Allerton Conference on Communications, Control and Computing*, 2006.
- [11] F. Mattoussi, V. Roca, and B. Sayadi, "Design of Small Rate, Close to Ideal, GLDPC-Staircase AL-FEC Codes for the Erasure Channel," in *IEEE Global Communications Conference (GLOBECOM'12)*, Dec. 2012.
- [12] V. Roca, C. Neumann, and D. Furodet, "Low density parity check (ldpc) staircase and triangle forward error correction (fec) schemes," *IETF Request for Comments*, RFC 5170, Jun. 2008.
- [13] 3GPP, "3rd generation partnership project; technical specification group services and system aspects; multimedia broadcast/multicast service; protocols and codecs (release 6)," *3GPP TS 26.346 v1.5.0*, 2004.
- [14] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft, "Asynchronous layered coding (alc) protocol instantiation," *IETF Request for Comment*, RFC 3450, 2002.
- [15] T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh, "Flute - file delivery over unidirectional transport," *IETF Request for Comments*, RFC 3926, Oct. 2004.
- [16] V. Roca, M. Cunche, C. Thienot, J. Detchart, and J. Lacan, "RS + LDPC-Staircase Codes for the Erasure Channel: Standards, Usage and Performance," in *9th IEEE Int. Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'13)*, Oct. 2013.
- [17] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," *IETF Request for Comments*, RFC 5053, Oct. 2007.
- [18] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery," *IETF Request for Comments*, RFC 6330, Aug. 2011.



**Fig. 2:** Total number of successful decodings under IT+RS (left) or ML (right) decoding, with respect to the ABL, with PLR=50% (left) or 66.66% (right), when  $r_G = \frac{1}{3}$ .

[19] E. N. Gilbert, "Capacity of burst-noise channel," *Bell System. Technical Journal*, vol. 39, p. 12531266, 1960.

[20] J. Bolot and A. V. Garcia, "The case for fec-based error control for packet audio in the internet," *ACM Int. Conference of Multimedia*, 1996.

[21] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss runlength," *SPIE/ACM SIGMM Multimedia Computing Network Conference*, 2000.

[22] C. C. Tan and N. C. Beaulieu, "On first-order markov modeling for the rayleigh fading channel," *IEEE Transaction on Communications*, vol. 48, pp. 2032–2040, 2000.

[23] E.O.Elliot, "A model of the switched telephone network for data communications," *Bell system Technical Journal*, vol. 44, pp. 89–109, 1963.

[24] F. Mattoussi, V. Roca, and B. Sayadi, "Optimization with EXIT functions of GLDPC-Staircase codes for the BEC," in *IEEE 13th Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC'12)*, Jun. 2012.

[25] F. Mattoussi, B. Sayadi, and V. Roca, "Good Coupling Between LDPC-Staircase and Reed-Solomon for the Design of GLDPC Codes for the Erasure Channel," in *IEEE Wireless Communications and Networking Conference (WCNC'13)*, Jan. 2013.