Analysis of FEC Codes for Partially Reliable Media Broadcasting Schemes

Christoph NEUMANN and Vincent ROCA

INRIA Rhône-Alpes, Planète Research Team, France[**]
{firstname.lastname}@inrialpes.fr

Abstract. With many multimedia delivery schemes a client does not necessarily receive a given media content entirely (e.g. router congestions can lead some information to be lost). Mechanisms like error concealment and error resilient video coding allow the receiver to deal with partially received data and to play the content, with decreased quality though. Packet-level Forward Error Correction (FEC) can be used as a complementary technique to counter the effects of losses and reconstruct the missing packets. However if the number of packets received is too low for the FEC decoding process to finish, the received parity packets may turn out to be useless, and finally more source packets may be unavailable to the application than if FEC had not been used at all. This paper analyzes the adequacy of the LDGM Staircase, LDGM Triangle and RSE FEC codes to offer a partial reliability service for media content distribution over any kind of packet erasure channel, that is to say a service that enables a receiver to reconstruct parts of the content even if the FEC decoding process has not finished. We analyze this service in the context of a broadcasting system having no feedback channel and that offers media content distribution, like Digital Video/Audio Broadcasting.

1 Introduction

This work analyzes a FEC-based partial reliability service in the context of a media delivery system like DVB or DAB (Digital Video/Audio Broadcasting). Data broadcasting to cars (e.g. [2]), video delivery over satellites, DVB-T/H using "IP Datacast" (IPDC) [8] are possible applications for such types of transmissions. Because there is no back channel, no repeat request mechanism can be used that would enable the source to adapt its transmission according to the feedback information sent by the receiver(s). The lack of feedback channel however enables an unlimited scalability in terms of number of receivers, who behave in a completely asynchronous way. Typically the media content is transmitted in a carousel (or similar) approach, i.e. packets are transmitted cyclically and/or randomly for a significant duration that usually exceeds the transmission time of a single copy of the content. Using a reliable multicast transmission protocol like ALC [4] (which can in fact offer either a fully or partially reliable delivery

---

service, depending on the way it is used), along with the FLUTE [9] file delivery application, can turn out to be highly effective in this context [2].

Yet, in order to be efficient, these approaches largely rely on the use of a Forward Error Correction (FEC) scheme. After an FEC encoding of the content, redundant data is transmitted along with the original data. Thanks to this redundancy, up to a certain number of missing packets can be recovered at the receiver. More precisely k *source packets* (A.K.A. original or data packets) are encoded into n packets (A.K.A. encoding packets). If the FEC encoder keeps the k source packets in the set of n packets, this code is called *systematic*, which is the case of all three FEC codes considered here. The additional $n - k$ packets are called *parity packets* (A.K.A. FEC or redundant packets). A receiver can then recover the k source packets provided it receives any k (or a little bit more with LDGM/LDPC codes) packets out of the n possible. The great advantage of using FEC with multicast or broadcast transmissions is that the same parity packet can recover different lost packets at different receivers.

The FEC decoding process may not end successfully if not enough packets have been received by the receiver (i.e. less than k or a little bit more). This is the case when reception is interrupted (e.g. a carousel-based transmission may be active only for a limited duration) or is affected by a too important amount of losses (e.g. if a vehicle enters an environment with many obstacles that only enables an erratic connectivity). A partial reliability service should maximize the amount of original content reconstructed at a receiver, even if FEC decoding did not finish. The target of this work is therefore to *analyze the inherent capabilities of each FEC code to maximize this partial decoding, and to find the best operational conditions to achieve this goal.*

The paper is structured as follows: Section 2 gives an introduction to RSE and LDGM codes. Section 3 presents the issues with a partial reliability service for RSE, LDGM Staircase and LDGM Triangle. Sections 4 and 5 introduce experimental results in specific scenarios. Finally we conclude.

## 2 Introduction to RSE, LDGM Staircase and LDGM Triangle codes

2.1 RSE Code

Reed-Solomon erasure code (RSE) is one of the most popular FEC codes. RSE is intrinsically limited by the Galois Field it uses [10]. A typical example is $GF(2^8)$ where $n \leq 256$. With one kilobyte packets, a FEC codec producing as many parity packets as data packets (i.e. $n = 2k$) operates on blocks of size 128 kilobytes at most, and all files exceeding this threshold must be segmented into several blocks, which reduces the global packet erasure recovery efficiency (e.g. if B blocks are required, a given parity packet has a probability $1/B$ to recover a given erasure, and $B = 1$ is then the optimal solution). This phenomenon is known as the "Coupon Collector Problem" [1]. Another drawback is a huge encoding/decoding time with large (k, n) values, which is the reason why $GF(2^8)$

is preferred to $GF(2^{16})$ in spite of its limitations on the block size. Yet RSE is optimal (we say it is an MDS code) because a receiver can recover erasures as soon as it has received *exactly* k packets out of n.

2.2 LDGM Codes

We now consider another class of FEC codes that completely departs from RSE: Low Density Generator Matrix (LDGM) codes, that are variants of the well known LDPC codes introduced by Gallager in the 1960s [3].

(a) Bipartite graph   (b) H matrix
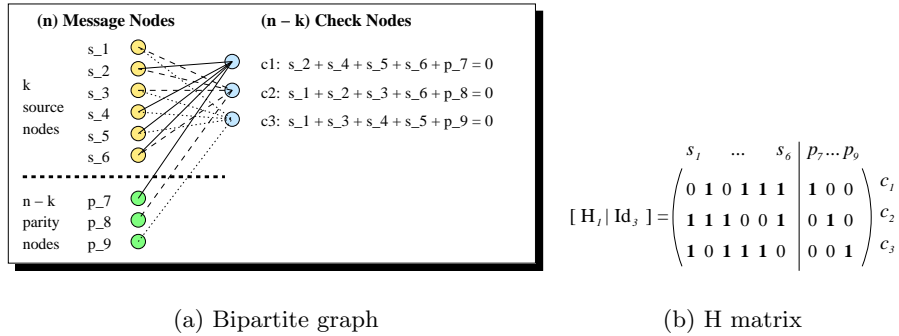
Fig. 1. A regular bipartite graph and its associated parity check matrix for LDGM.

Principles: LDGM codes rely on a bipartite graph between left nodes, called *message nodes*, and right nodes, called *check nodes* (A.K.A. constraint nodes). The k source packets form the first k message nodes, while the parity packets form the remaining $n - k$ message nodes. The upper part of this graph is built following an appropriate left and right degree distribution (in our work the left degree is 3). The lower part of this graph follows other rules that depend on the variant of LDGM considered (e.g. with LDGM, figure 1 (a), there is a bijection between parity and check nodes). This graph creates a system of $n - k$ linear equations (one per check node) of n variables (source and parity packets).

A dual representation consists in building a parity check matrix, H. With LDGM, this matrix is the concatenation of matrix H_1 and an identity matrix I_{n-k}. There is a 1 in the $\{i; j\}$ entry of matrix H each time there is an edge between message node j and check node i in the associated bipartite graph.

Thanks to this structure, *parity packet creation is straightforward and extremely fast*: each parity packet is equal to the sum of all source packets in the associated equation. For instance, packet p_7 is equal to the sum: $s_2 \oplus s_4 \oplus s_5 \oplus s_6$. Besides LDPC/LDGM codes can operate on *very large blocks*: several tens of

megabytes are common sizes. However LDGM is not an MDS code and it introduces a decoding inefficiency: $inef\_ratio * k$ packets, with $inef\_ratio \geq 1$, must be received for decoding to be successful. The $inef\_ratio$, experimentally evaluated, is therefore a key performance metric.

Iterative Decoding Algorithm: With LDGM, there is no way to know in advance how many packets must be received before decoding is successful (LDGM is not an MDS code). *Decoding is performed step by step, after each packet arrival, and may be stopped at any time.*

The algorithm is simple: we have a set of $n-k$ linear equations of n variables (source and parity packets). As such this system cannot be solved and we need to receive packets from the network. Each non duplicated incoming packet contains the value of the associated variable, so we replace this variable in all linear equations in which it appears. If one of the equations has only one remaining unknown variable, then its value is that of the constant term. We then replace this variable by its value in all remaining equations and reiterate, recursively. As we approach the end of decoding, incoming packets tend to trigger the decoding of several packets, until all of the k source packets have been recovered.

LDGM Staircase Code: This trivial variant, suggested in [5], only differs from LDGM by the fact that the I_{n-k} matrix is replaced by a "staircase matrix" of the same size. This small variation affects neither encoding, which remains a simple and highly efficient process, nor decoding, which follows the same algorithm. But this simple variation *largely improves the decoding efficiency.*

LDGM Triangle Code: This is a variant of LDGM Staircase where the empty triangle beneath the staircase diagonal is now filled, following an appropriate rule [12]. This rule adds a "progressive" dependency between check nodes, as shown in figure 2. This variation further increases performance in some situations, while keeping encoding highly efficient (even if a bit slower since there are more "1"s per row). Here also decoding follows the same iterative algorithm.

Interested readers are invited to refer to [11, 12]. A publicly available, open source, implementation of the codes is also available at [7].

3 Partial Reliability: the Solution Space

We now explain the various possibilities when designing a partial reliability service.

3.1 Self Sufficiency versus Interdependency

A video content is composed of I, P and B video frames, and an audio content is composed of samples also called frames. All audio frames and the I video frames
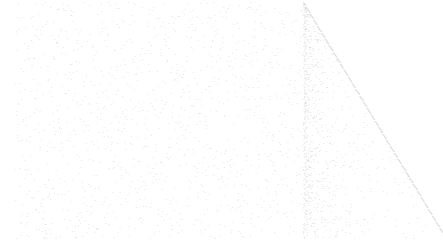
Fig. 2. Parity check matrix (H) for LDGM Triangle (k=400, n=600).

are self-sufficient entities, and the transmission scheme must absolutely preserve this self sufficiency property[1].

But P video frames depend on the previous I and P frames, and B video frames depend on the previous/next I and P frames. So in a partially reliable transmission scheme, there is an incentive to reconstruct the self sufficient frames first, before reconstructing the entire content. It leads to the idea of Unequal Error Protection (UEP) that will be considered in future works.

3.2 Object Definition

We call *object* an entity defined by the application and submitted to the transport layer as a whole. This definition is rather generic and therefore objects may largely differ, depending on the application: it can be an entire file, an audio track, or just a frame. If a large block LDGM-* FEC codec is used in the transport layer, the entire object, even if large, will probably be encoded as a whole (single block). But with an RSE codec, a large object has to be segmented into several blocks. As we'll see later on, this major difference will largely influence the efficiency of the partial reliability service.

3.3 Frame to Object Mapping

Several solutions exist to apply the notion of object to a media content, and control how FEC encoding is performed:

1- Map each frame to a separate object and encode them independently: the application can exploit a frame as soon as the corresponding object has been decoded (and all the other frames it may depend on with P and B video frames). Even if the frame size varies, depending on the bit rate and frame type, this size remains small, from several hundreds to a few thousands of bytes. Therefore there is no object segmentation, even with RSE. But LDGM-* codes have poor performance when operating on small block sizes, unlike RSE which benefits from its MDS feature [12]. So we can expect bad results with LDGM codes here. Additionally, since there is a high number of objects, all codes will

---

[1] This is the same requirement as when defining an RTP framing scheme with some multimedia content.

suffer from the Coupon Collector Problem (section 2.1). So we can expect all codes to have bad performance.
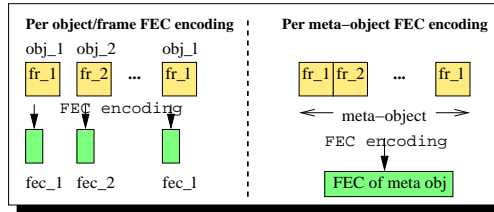
Fig. 3. Per object versus per meta-object FEC encoding.

2- Map the entire content to a single "meta object" and encode it: encoding is now performed over the whole meta-object for LDGM-*, and over blocks of maximum size for RSE. So the Coupon Collector Problem is totally eliminated with LDGM-* codes, and largely reduced with RSE (there are fewer blocks). Besides we know that LDGM-* codes perform very well with large objects [12], so their use seems rather promising.

Since a receiver must be able to reconstruct individual frames of a partially received meta-object, the underlying structure of the meta-object must be communicated. This can be achieved within the transport protocol by means of a dedicated signaling header extension (e.g. we added a private ALC header extension that contains a description of the meta-object and that is sent periodically).

Finally, operating on meta-objects means that FEC encoding is done over several seconds or minutes of the media content, or even over the entire media file. A significant latency is thus introduced in the transmission model, which is an issue in case of real-time streaming, but is not a problem with the test cases considered in this paper, or in our SVSoA proposal [6].

3.4   Benefits of an Iterative Decoding System: An intuition

Intuitively, an iterative decoding algorithm should have a major advantage over MDS codes in partially reliable sessions. The reason is that decoding is done progressively and can be stopped at any time. A receiver can therefore exploit the subset of source packets received *plus* the subset of source packets already decoded. On the opposite, decoding with RSE is only possible once exactly k distinct packets have been received. If the session stops before this threshold, then no decoding is possible, and only the subset of source packets in the packets received can be exploited.

This fundamental difference is intrinsic to the FEC codec. However the intuitive conclusion that LDGM codes are more suited to a partial reliability service is not always true, as we will show later on.

4 One-to-One Frame to Object Mapping

4.1 Experimental Conditions

All the experiments use our LDPC/LDGM codec [7] and a popular RSE codec [10]. We designed a basic application on top of these two codecs, derived from the `perf_tool` of our FEC codec. Given one or more objects, the application first performs FEC encoding: to k source packets of a block, the codec produces k parity packets, all packets being 512 bytes long. Then all the source and parity packets (of all blocks) are transmitted in a fully random manner, indefinitely, by choosing one packet out of all the possibilities (to mimic a random carousel transmission). The receiving application waits until it has received enough packets to decode the block (or all blocks with RSE) and stops.

As can be noticed, there is no explicit loss model or loss rate, because random transmissions are not affected by them. Performance results could be slightly different if other transmission models were used, for instance by sending all source packets in sequence except a few of them, assumed erased, and then parity packets. But in that case we introduce additional parameters like the loss model and the loss ratios. In contrast our approach avoids these problems, is more universal, simulates lossy transmissions with a random carousel like scheme, and is in line with transmissions like data broadcasting to cars [2], DVB IP Datacast [8], or a video streaming approach like [6].

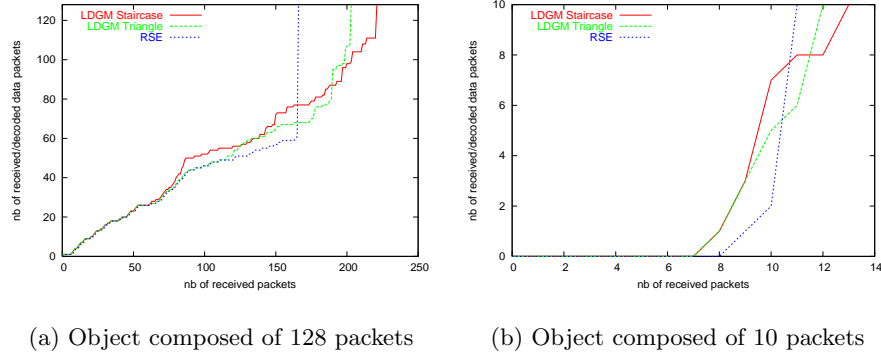## 4.2 Performance With a Single Small Object

This first experiment illustrates the decoding behavior of each code. We consider a small object that fits into one RSE block (i.e. 128 source packets and 128 parity packets), and test all three codes using exactly the same packet sequence. Results are shown in figure 4 (a).

Since RSE is an MDS code, it decodes the object first, after receiving 166 packets (this value is ≥ 128 because of duplicated packet reception made possible by the random carousel transmission scheme). But RSE does not offer any partial reliability service in this scenario, and the source packets available earlier are those that have been received. The LDGM-* codes need more packets to decode the entire object. Yet source packets start to be decoded sooner, after 63 receptions for LDGM Staircase and 116 for LDGM Triangle. We also see that the Triangle version finishes the entire decoding sooner, but starts partial decoding after the Staircase version (because of extra relationships in the H matrix between all packets).

We did the same with a smaller object of size 10 packets, closer to an actual audio or video frame size. Figure 4 (b) shows the same general behavior as previously, at a smaller scale.

4.3 Performance With a Large Number of Small Objects

Here we transmit 600 objects of size 10 packets each, for a total content of about 3 MBytes, or 6000 packets of size 512 bytes. Results are shown in figure 5.

(a) Object composed of 128 packets   (b) Object composed of 10 packets

Fig. 4. Number of decoded packets with a single small object.
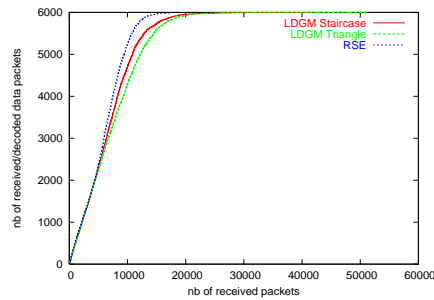
Fig. 5. Number of decoded packets with a per object encoding and a large number of small objects.

As expected, the general performance behavior is rather bad with all codes, especially with LDGM codes: RSE needs 17651 packets, LDGM Staircase 41677, and LDGM Triangle 50971. These poor results, caused by the coupon collector problem and a sub optimal use of LDGM codes, are not acceptable.

5 Frame Aggregation in a Single Meta Object

5.1 Performance Without Inter-Frame Dependencies

Single Packet Frames: We first analyze each code in case of a meta object of size 6000 packets, assuming each frame forms a single packet. Results are reported in figure 6. We see LDGM Triangle finishes decoding the first (after 9403 packets received), followed by LDGM Staircase (after 9792 packets received) and then RSE (after 10282 packets received).

But partial reliability requires that decoding starts as soon as possible. LDGM Staircase starts decoding some data packets the first, after 798 received packets,
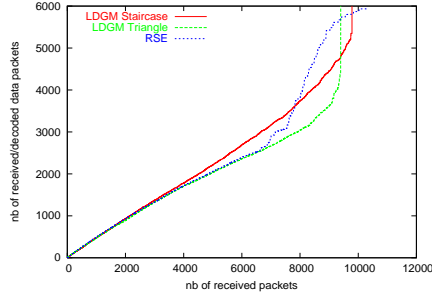
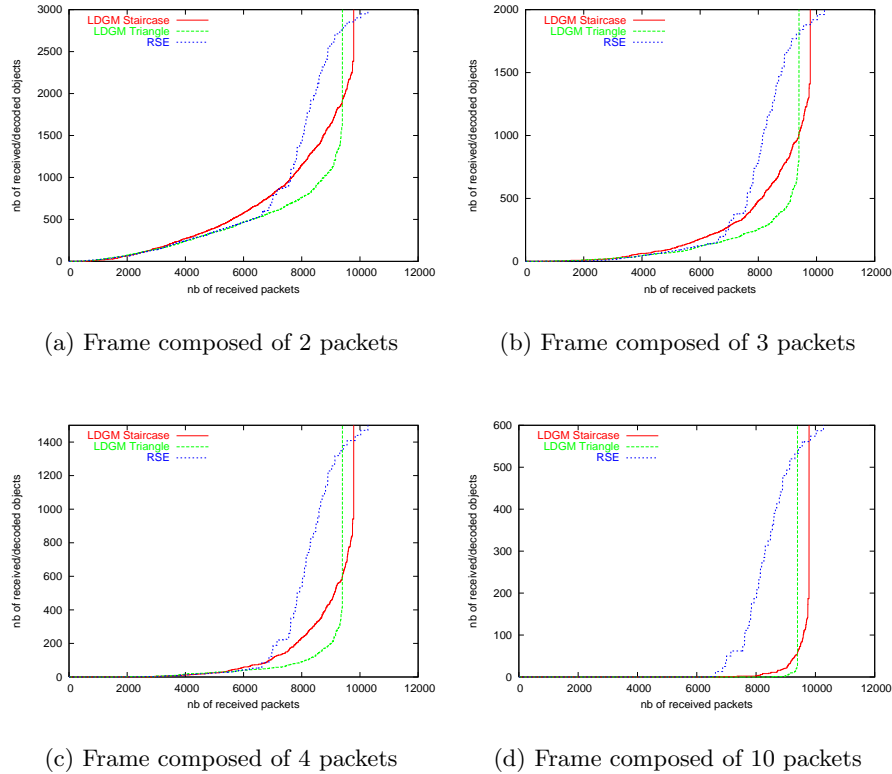**Fig. 6.** Number of decoded packets of a meta-object.

but it is clearly visible only after ≈ 3000 packets. At this point LDGM Staircase offers the best partial reliability service. RSE begins to decode the first block after 6634 received packets, and then quickly crosses LDGM Staircase after ≈ 7800 packets. At this point RSE offers the best partial reliability service. LDGM Triangle performs quite poorly since decoding is essentially done at the end. No partial reliability service is offered by this code, even if it exhibits the best decoding performance of all three codes!

Frames That Span Several Packets: We now consider a more realistic situation where frames span several packets. We analyzed the number of decoded frames as a function of the number of received packets. To simplify, we assumed that all the frames have the same size. If the object size is 1 packet, then the previous results of figure 6 can be used. In figure 7 we reported the results for object sizes of 2,3,4 and 10 packets respectively. We clearly see that RSE offers in all cases the best partial reliability service. The RSE performance even becomes better compared to the LDGM codes as the frame size increases.

To better understand it, we also analyzed the time at which a given packet is decoded. Results are shown in figures 8. We clearly see that with RSE, packets are decoded per block, making a group of adjacent packets available to the application at the time of the decoding. On the opposite, both LDGM codes make the packets available in a completely flat and random manner. This is an issue if a frame spans several packets, since one of them may not be available. *Therefore RSE offers a better partial reliability service than LDGM codes when frames span several packets.*

5.2 Performance With Inter-Frame Dependencies

Finally we did similar tests but with an `IPPPPI` sequence of video frames, where `I` and `P` frames are respectively 10 and 5 packets long. Remind that a `P` frame depends on the availability of the previous `I` (and `P` if any) frame(s) of the current group of frames. The whole sequence is composed of 1000 frames, among which 200 are `I` frames. This test is therefore extremely close to an actual video media.

(a) Frame composed of 2 packets

(b) Frame composed of 3 packets

(c) Frame composed of 4 packets

(d) Frame composed of 10 packets

Fig. 7. Number of decoded frames of a meta-object.

Results, shown in figure 9, are rather close to that of figure 7. Here also RSE is clearly the best solution to offer a partial reliability service.

6 Conclusions and Future Works

We have analyzed the ability of three FEC codes – LDGM Triangle, LDGM Staircase and RSE – to offer a partial reliability service in case of media broadcasting schemes, that largely differ from more classical streaming approaches. We also introduced a meta-object encoding that largely improves the global efficiency of the broadcasting system. This analysis is quite innovative and we are not aware of any similar work. Traditionally only the global inefficiency ratio, which indicates how many packets must be received for decoding to complete, was considered. Our most important result is that LDGM-* codes, in spite of their iterative decoding approach and high efficiency, are not appropriate when partial reliability is desired. In contrast, RSE codes can more efficiently cope

(a) RSE

(b) LDGM Staircase

(c) LDGM Triangle
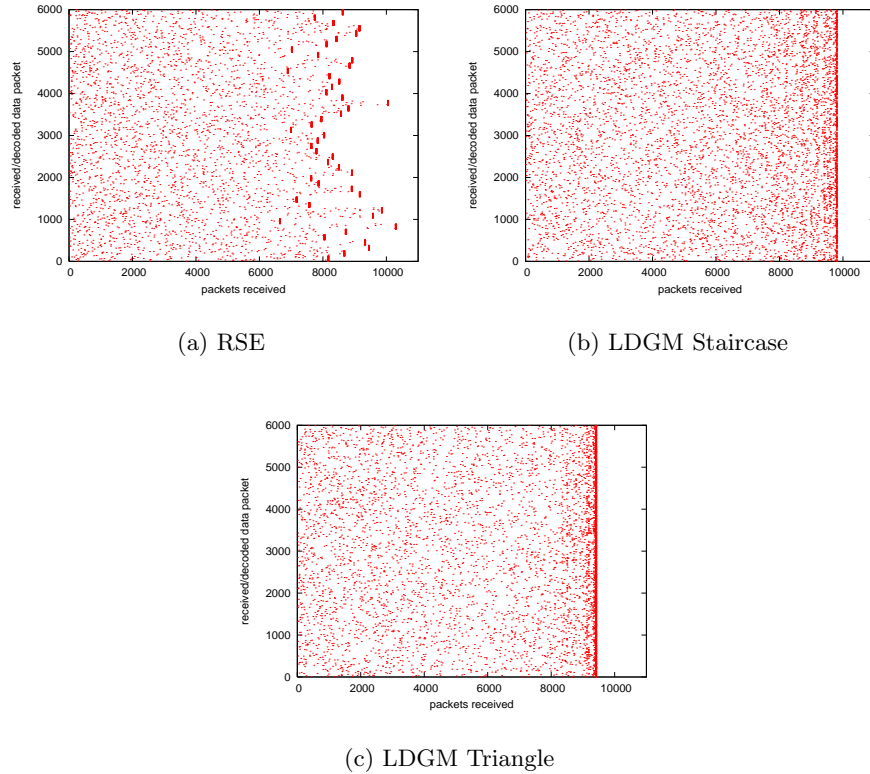
Fig. 8. Repartition of the received/decoded source packets of a meta-object as a function of time.

with this requirement. This is rather counter-intuitive and only detailed experimentations and analyzes led us to come to this conclusion.

Finally, these results will be directly used in our Scalable Video Streaming over ALC (SVSoA) proposal [6]. Future works will also consider Unequal Error Protection FEC schemes, that should prove to be interesting to cope with inter-frame dependencies.

References

1. J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *ACM SIGCOMM'98*, Aug. 1998.
2. H. Ernst, L. Sartorello, and S. Scalise. Transport layer coding for the land mobile satellite channel. In *59th IEEE Vehicular Technology Conference (VTC'04), Milan, Italy*, May 2004.
3. R. G. Gallager. Low density parity check codes. *IEEE Transactions on Information Theory*, 8(1), Jan. 1962.
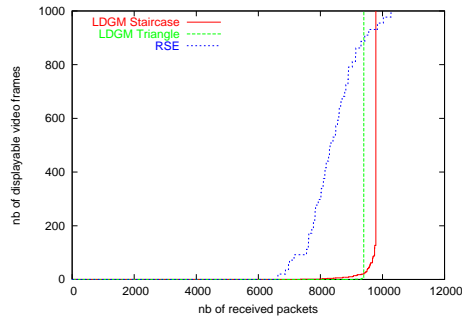
Fig. 9. Number of displayable video frames of a meta-object with an IPPPPI sequence.

4. M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft. *Asynchronous Layered Coding (ALC) protocol instantiation*, Dec. 2002. IETF Request for Comments, RFC3450.

5. D. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, ISBN: 0521642981, 2003.

6. C. Neuman and V. Roca. Scalable video streaming over alc (svsoa): a solution for the large scale multicast distribution of videos. In *Streaming media distribution over the Internet (SMDI04), Athens, Greece*, May 2004.

7. C. Neumann, V. Roca, J. Labouré, and Z. Khallouf. *An Open-Source Implementation of a LDPC/LDGM Large Block FEC Codec.* URL: http://www.inrialpes.fr/planete/people/roca/mcl/.

8. T. Paila. Mobile internet over ip data broadcast. In *10th IEEE Int. Conference on Telecommunications (ICT'03), Papeete, French Polynesia*, Jan. 2003.

9. T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh. *FLUTE - File Delivery over Unidirectional Transport*, Dec. 2003. Work in Progress: <draft-ietf-rmt-flute-07.txt>.

10. L. Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM Computer Communication Review*, 27(2), Apr. 1997.

11. V. Roca, Z. Khallouf, and J. Laboure. Design and evaluation of a low density generator matrix (ldgm) large block fec codec. In *Fifth International Workshop on Networked Group Communication (NGC'03), Munich, Germany*, Sept. 2003.

12. V. Roca and C. Neumann. Design, evaluation and comparision of four large block fec codes, ldpc, ldgm, ldgm staircase and ldgm triangle, plus a reed-solomon small block fec codec. Research Report 5225, INRIA, June 2004.