Integration of small and large packet-level FEC codes with data carousels for reliable multicast transport over satellite links

M. Chipeta, L. Fan, B.G. Evans
Mobile Communications Research Group
CCSR, University of Surrey
Guildford, GU2 7XH, UK
m.chipeta@surrey.ac.uk

M. Karaliopoulos
New Technologies and Applications Group
Teletel SA
116 36 Athens, Greece
M.Karaliopoulos@teletel.gr

***Abstract*—Automatic repeat request (ARQ), data carousels and packet-level forward error correction (PLFEC) are fundamental techniques for combating packet loss in point-to-multipoint communications. Generally speaking, ARQ-based techniques are unattractive due to the feedback implosion they generate. Furthermore, in broadcast/multicast wireless networks, radio resources are scarce and feedback may not be an option at all. Two error mitigation mechanisms that become relevant in this scenario are data carousels and PLFEC. In this work, we consider the application of these two techniques as components of an integrated layer in satellite environments. We also investigate the impact of the type of PLFEC code, small or large, on the main user-oriented performance metric of the data carousels, the average content download time.**

Index Terms*—*Data carousels; FEC; point-to-multipoint communications; reliable multicast

I. INTRODUCTION

One of the key research issues concerning broadcasting to mobile users over wireless links is reliable content delivery. Information loss may occur for a number of reasons: radio transmission errors due to the impairments of the wireless link(s), intermittent connectivity during cell handovers in cellular networks and more system-specific reasons such as the occasional preemption of the satellite signal reception by the terrestrial cellular mobile network in the case of the forthcoming Satellite Digital Multimedia Broadcasting (SDMB) system [1]. Data loss may be tolerable to a certain extent for some applications, e.g., video, but not for other applications such as software distribution. Although data loss mitigation techniques are deployed at the access layer of mobile wireless networks, such as forward error correction, interleaving, interference cancellation, and applications feature their own mechanisms for coping with data loss, such as error concealment techniques for multimedia applications, additional protection at the transport layer is deemed necessary in order to satisfy the performance requirements of the envisaged applications.

In wired networks, the feedback implosion effect related to the automatic repeat request (ARQ)-based techniques is addressed via use of back-off timers and feedback aggregation; these approaches take advantage of the tree structure of wired networks and the variable delay therein. The same techniques are more difficult to implement in wireless access networks, in particular satellite systems, which serve a much higher population of terminals and often feature strict power budgets on both the satellite and the satellite terminal side. These same reasons may even completely preclude the use of a feedback channel. Packet-level forward error correction (PLFEC) and data carousels, possibly combined with interleaving, are then the remaining options for the design of the reliable multicast transport layer. Notably, three emerging solutions for multimedia broadcasting to mobile terminals, the Third Generation Partnership Project (3GPP) Multimedia Broadcast/Multicast Service (MBMS) framework [2], the European Digital Video Broadcasting standard for mobile handheld terminals (DVB-H) [3], and the SDMB system, rely on some form of PLFEC, hereafter called FEC, and data carousels for enhancing the reliability of data delivery, all avoiding the use of ARQ-based schemes.

Both error mitigation techniques have been studied extensively but separately. We first proposed an integrated carousel-FEC layer employing small FEC codes in [4]. Herein, we extend that work in investigating the use of large codes and the combination of both small and large codes in the implementation of data carousels, focusing on satellite environments. We first describe the considered satellite system model in section II, before outlining the integrated carousel-FEC layer in section III and the main analytical equations related to it. Numerical results are presented and discussed in section IV and our conclusions are given in section V.

II. THE SATELLITE LINK MODEL

The satellite network setting we consider is illustrated in Fig. 1. Despite ignoring the specific engineering details of different systems, the model serves the analytical objectives of this paper.

The elementary data unit we consider is the UDP/IP packet. Generally, the end-to-end communication may suffer data loss

This work has partially been supported by the European Union in the context of the Framework Program 6 Integrated Project MAESTRO (Mobile Applications & sErvices based on Satellite and Terrestrial inteRwOrking). MAESTRO started in January 2004 and its first phase is planned to finish in December 2005.

Website: http://ist-maestro.dyndns.org/MAESTRO/index.htm.

0-7803-9206-X/05/$20.00 ©2005 IEEE

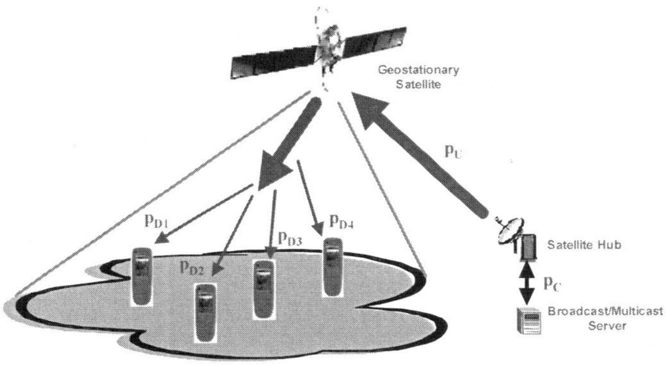

Figure 1. Simple model of the via-satellite content path

due to either congestion at the wired path segment (sender-satellite gateway) or link impairments over the wireless link.

Let p_C be the mean packet loss probability at the wired segment. On the other hand, radio link impairments may give rise to either uniform or burst packet loss at the packet level, depending on the propagation channel dynamics, the user mobility and the specific access layer mechanisms. In this paper, we adopt the assumption of uniform packet loss, leaving the study of burst packet loss impact for future work. We discriminate between the mean packet loss probability at the uplink p_U and packet probability loss at the downlink p_D. Note that p_U is the same for all sender-receiver pairs, whereas p_D varies due to the spatial separation and/or qualitative differentiation of receivers.

The overall probability of packet loss, assuming independence of the packet loss processes in the three segments, is

$$p = 1 - (1 - p_U) \cdot (1 - p_D) \cdot (1 - p_C). \tag{1}$$

Assuming that packet loss at the wired segment is negligible, $p_C = 0$, the end-to-end packet loss can be written

$$p = p_U + p_D - p_U \cdot p_D \tag{2}$$

III. The Integrated Carousel-FEC Layer

With data carousels information is organized into data items, corresponding to a single file or a batch of files, which are transmitted repeatedly in the broadcast medium according to a specific schedule. Users have the chance to acquire items of interest to them only at given time instants corresponding to the occurrences of these items in the schedule, rather than on demand. In case they do not succeed in retrieving the whole item by the i^{th} attempt, they will have to wait for subsequent appearances of the item till they retrieve it correctly. However, a well-designed schedule takes into account the relative demand for each data item, so that the number of appearances of each item in the schedule increases with its demand probability. This way the average time required from a random user to retrieve an item can be minimized [5] and the interactivity perceived by the user *(pseudo-interactivity)* improves.

Let t_k^i be the *download* time for item i and user k; namely, the time that elapses between the time instant when the user expresses his desire to access item i, to the time when the item is retrieved from its schedule and is stored at his terminal [5]. The mean response (or download) time S_r, the key user-oriented metric for the carousel efficiency, is then defined as the expected value of t_k^i when considering the whole user population and all carousel items. It is shown in [5] that under the optimum broadcast schedule design strategy, the average response time S_{opt} may be written as

$$S_{opt} = \frac{1}{2} \cdot \left(\sum_{i=1}^{M} \sqrt{q_i \cdot l_i \cdot \left\lceil 1 + 2r_i(p) \right\rceil} \right)^2, \tag{3}$$

where M is the number of items in the schedule and the function r_i computes the mean number of required re-appearances of data item i (each one associated with a demand probability value q_i and length l_i) after its first appearance, so that it is fully retrieved in the presence of data loss.

In the integrated carousel-FEC layer, the individual items that compose the data carousel are encoded. Coding is applied at the transport layer, resulting in additional packets being transmitted on top of the original data packets. The information redundancy that is added to the original data aims at quicker item recovery when data loss occurs.

The joint use of FEC and carousels introduces a trade-off with respect to the mean download time. Higher levels of FEC redundancy improve the error correction capability of the FEC codes, resulting in quicker download times, since a user has higher chances to download an item in fewer attempts. On the contrary, higher FEC redundancy levels add to the equivalent length of items and increase the spacing between appearances of a particular item in the schedule, resulting in higher response times. An integrated carousel-FEC design is characterized optimum, when it minimizes the average user download time, whilst also keeping the redundancy introduced by FEC at a minimum.

A. Configuration alternatives for the integrated carousel-FEC layer

The integrated carousel-FEC layer may be configured with respect to the following aspects:

1) Item Retrieval Technique (IRT): The way the data items are retrieved from the transmitted carousel may vary: the 1-shot retrieval and the cumulative retrieval are the two main alternatives. With 1-shot retrieval, the application attempts to recover an item from the carousel in one-go; therefore, if an item is recovered partially, the correctly received packets are discarded and the retrieval of the item in question starts from scratch upon its next appearance. On the other hand, the cumulative approach entails the storing of correctly received packets and the step-by-step item retrieval during its successive appearances.

2) Type of FEC code (large or small): In general, there are two types of codes: small and large [6]. Small codes e.g., Reed-Solomon (RS), require larger files to be split up into several FEC blocks in order to better manage the complex Galois Field arithmetic. In contrast, large codes e.g., Low Density Generator

Matrix (LDGM), require simple XOR operations; thus, large codes have higher codec throughputs compared to small codes [6], [7]. The capability of large codes to encode a large file in one FEC block is quite beneficial since large FEC blocks have higher bandwidth efficiency compared to small FEC blocks as demonstrated in [6], [7], and section IV herein. The advantage of large codes with respect to the achievable codec throughput has to be assessed taking into account the transmission rates supported by the different systems. For low transmission rates, e.g., the upper limit of 384kbps for MBMS and S-DMB, the system bottleneck will be the transmission capacity rather than the codec speed. On the other hand, benefits are more evident in the case of low-end handheld devices with limited processing resources.

The advantage of small codes is that in order to recover a FEC block, their codec requires the correct reception of at least any k packets out of the n sent as indicated by (4). On the contrary, a large decoder requires a minimum number of $(1+r_o)\cdot k$ packets to recover the original k, where r_o is the reception overhead which is lower for large files compared to smaller ones. This reception overhead[1] exists for large codes because, unlike in a RS FEC block, each equation generating each parity packet does not involve all the original packets.

The corresponding equations for the small (large) code block error rate $FBLER^S$ ($FBLER^L$) and the probabilities of full acquisition of each item in the two cases $P_{F,100}^{S(L)}$ are given by

$$FBLER^S = \sum_{j=n-k+1}^{n} \binom{n}{j} p^j (1-p)^{n-j}. \tag{4}$$

$$P_{F,100}^S = \left(1 - FBLER^S\right)^{N_B}. \tag{5}$$

$$FBLER^L = \sum_{j=n-k(1+r_o)+1}^{n} \binom{n}{j} p^j (1-p)^{n-j} = P_{F,100}^L. \tag{6}$$

We could generalize (3) and write for the mean download time with the integrated carousel-FEC layer:

$$S_{opt}(\underline{k},\underline{n}) = \frac{1}{2}\cdot\left(\sum_{i=1}^{M}\sqrt{q_i \cdot SF_i \cdot l_i \cdot \left[1 + 2r_i\left(k_i, n_i, N_B^i, r_o\right)\right]}\right)^2, \tag{7}$$

which, for the number of FEC blocks $N_B^i = 1$ and $r_o \neq 0$ points to large codes, whereas for $N_B^i \geq 1$ and $r_o = 0$, it captures the small codes. In both cases, the overhead due to the inclusion of parity packets is measured in terms of the stretch factor, *SF*, which is the ratio *n/k*. The reader is referred to [4] for options available with regards to the function r_i.

3) FEC overhead assignment rule: From the FEC perspective, we also have a choice with respect to how FEC redundancy is determined for each item in the carousel. We refer to the followed rule as a FEC assignment rule (FAR). In [4], we considered small codes and two FARs: fixed *SF* and fixed item success rate (see Fig. 2). In fixed *SF*, the same *SF* is applied to each item regardless of its size; in fixed item success

[1] A large, proprietary code known as Raptor, with an exceptionally low r_o, does exist and this has been adopted for 3GPP MBMS [2]. In this study, however, we limit our discussion to unrestricted codes such as LDGM codes.

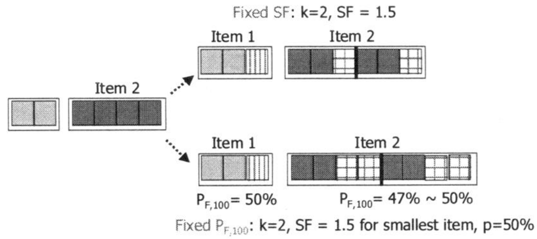

Figure 2. Two FARs based on small codes

rate, a target $P_{F,100}$ is chosen, based on the smallest item, leading to different *SF* values for items of different size.

IV. Numerical Results

A. Analytical investigation scenarios

1) FEC code use: We compare the performance of the integrated carousel-FEC layer under three scenarios with respect to the FEC code in use:

- Small FEC codes for every item *i*: $P_{F,100}^i = P_{F,100}^S, \forall i$
- Large FEC codes for every item *i*: $P_{F,100}^i = P_{F,100}^L, \forall i$
- Combined use of small and large codes depending on the length l_i of item *i*:

$$\textbf{if } l_i \geq l_{thr}\ P_{F,100}^i = P_{F,100}^L,$$

$$\textbf{else } P_{F,100}^i = P_{F,100}^S,$$

As mentioned in section III, r_o varies with file size. In the analysis herein, this variation is modeled by using the distribution given in Table I, which is an approximation of the simulation results for one of the LDGM variants presented in [7]. It is also assumed that the r_o values, as listed in Table I, are constant for all valid values of *SF*.

An item with l_{thr} or more packets is fed into the large codec. The value l_{thr} corresponds to the file size, where both the small and large codecs require the same *SF* to achieve a given file success rate. Beyond l_{thr}, the large codec is always more efficient than the small one, since the increased error correcting capabilities resulting from encoding over a single data block, outweigh the overhead due to the non-zero r_o. The use of a scheme employing a small or large FEC codec, hereafter called a hybrid codec, necessitates the determination of l_{thr}. Table II and the plots in Fig. 3 illustrate the impact of p on the performance difference between small and large codes. At very low p, small codes outperform large codes over the full file size range considered in Fig. 3, rendering l_{thr} irrelevant.

TABLE I. The Impact of File Size on Reception Overhead

r (%)	10	7	6	6		6
L	50	100	150	200		500

TABLE II. The Impact of Packet Error Rate on L_{THR}

p (%)	0.1	1	5	10	20	40
$\mathbf{l_{thr}}$	> 500	> 500	200	150	100	92 (~100)

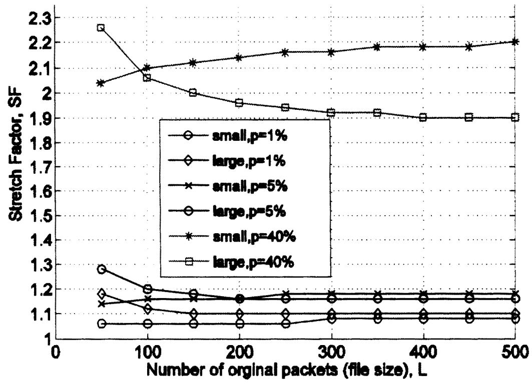

Figure 3. Peformance comparison between small and large codes to achieve $P_{F.100} \geq 99\%$; $k_{small} = 50$, $k_{large} = L$

TABLE III. DISTRIBUTIONS CONSIDERED IN THE ANALYSIS

Item demand probability	**Zipf**			
	$q_i = \frac{(1/i)^\theta}{\sum_i (1/i)^\theta}, 1 \le i \le M$			
	Distribution w.r.t. i	θ=0, uniform	θ=1, decreasing, lower slope	θ=2, decreasing, higher slope
Item length (FEC blocks)	**Uniformly variable**			
	$l_i = round\left[\left(\frac{z_1 - z_0}{M-1}\right)\cdot(i-1)+z_0\right]$			
	Increasing w.r.t. i if $z_1 > z_0$	Decreasing w.r.t i if $z_1 < z_0$	Random between z_1 and z_0 w.r.t i	

However, as p increases, large codes gradually outperform small codes, thus leading to l_{thr} values within the considered file size range, e.g., $l_{thr} = 200$ when $p = 5\%$. At high values of p, l_{thr} is smaller ($l_{thr} = 92$ when $p = 40\%$) and the performance gain of large codes over small codes is more significant. This behavior is in alignment with the simulation results in [8], which also indicate that as p grows, the efficiency gain of large codes over small codes increases while l_{thr} decreases.

2) Distributions for item demand and item length: the item demand is based on the Zipf function shown in Table III, where the parameter θ determines the nature of the distribution. An item is assigned its length and demand probability by superimposing the two distributions.

3) Item retrieval and FARs: in [4], the differences in performance between 1-shot and cumulative item retrieval as well as between fixed-SF and fixed item success rate FARs were shown to be minimal. In the following, the 1-shot item retrieval approach and the fixed SF FAR are used.

4) Packet loss distribution amongst users: we consider two scenarios with respect to packet loss distribution amongst the user population: homogeneous and heterogeneous.

B. Homogeneous loss scenario

In this scenario, we assumed a uniform probability of error p for all users, namely p_D is the same for all users. Fig. 4 depicts typical results of S_r versus *SF*: without FEC, the response time is very high; as *SF* grows, the response time becomes shorter, due to better error correction capability, until a minimum is reached. Beyond this minimum, additional redundancy is counterproductive since the effect of longer items (caused by increasing *SF*) dominates over the error correction capability. It is also interesting to note that at very low *SF* values, the large scheme is outperformed by the small one; this is a key characteristic of the two codes since at these *SF* values, the penalty induced by r_o is greater than the penalty due to splitting some items into a number of FEC blocks.

As listed in Table IV, at the optimum points the best hybrid scheme shows a 4% gain in *SF* and a lower response time by about 127 slots compared to the small scheme when $p = 10\%$. With respect to the large scheme, the hybrid approach has 2% gain in capacity and a lower response time by about 92 slots. In the settings used to plot Fig. 4, the item size varies uniformly in multiples of k (=50) from 50 to 500. Given $l_{thr} = 150$ at $p = 10\%$, the bulk of items in the carousel are classified as large; hence, the overall performance gain of the hybrid scheme over the large scheme is not as substantial as its gain over the small scheme. As can be seen from Table IV, wherever l_{thr} exists within range, the hybrid scheme always outperforms the other two schemes. Although, the hybrid scheme offers no advantage in capacity at high values of p, e.g. $p = 40\%$, it still manages to offer a lower response time compared to the large scheme. In the case where the majority of the items are small, i.e. at low values of p, the small codec outperforms the large codec and even the hybrid codec. The deduction of whether the majority of the items are small or large depends on p which determines l_{thr}. Furthermore, in the case where the majority of the items can be classified as large, the extent of the majority matters, i.e., it determines whether or not the small codec beats the large codec, because small codes have a substantial advantage on very small files.

For mobile satellite systems, typical packet loss rates can be in excess of 10%, which is the region where the hybrid scheme is particularly effective as shown by the results in Table IV. At these high packet loss rates, the hybrid scheme is also sensitive to the value of l_{thr}. If this threshold is carelessly chosen e.g. $l_{thr} = 300$ at $p = 40\%$ when the correct value given by the characterization is $l_{thr} = 100$, a significant rise in both the capacity consumed and the user response time can be observed. These results are consistent for other combinations of demand and length distributions, not reported here due to space limitations.

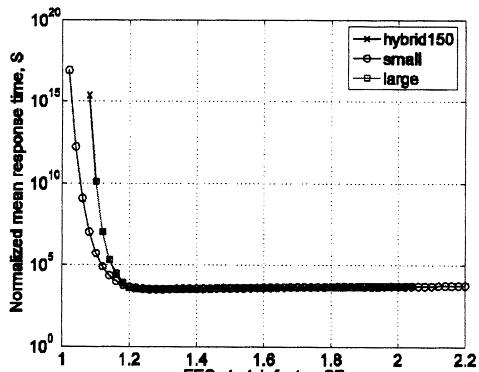

Figure 4. Comparison between three different codecs, including a hybrid shceme with $l_{thr} = 150$; $z_1 = 10$, $z_0 = 1$, $\theta = 0$, $p = 10\%$

TABLE IV. OPTIMAL POINTS FOR DIFFERENT CODECS AND PACKET ERROR RATES; $L_I = 10$, $\Theta = 0$ IN A HOMOGENEOUS LOSS SCENARIO.

codec		hybrid100		hybrid150		hybrid200		hybrid300		small		large	
l_{thr}	p (%)	S_{opt}	SF	S_{opt}	SF	S_{opt}	SF	S_{opt}	SF	S_{opt}	SF	S_{opt}	SF
> 500	0.1	2773	1.08	2771	1.08	2771	1.08	2771	1.08	2657	1.02	2829	1.10
> 500	1	2833	1.10	2823	1.10	2822	1.10	2822	1.10	2778	1.08	2890	1.12
200	5	3035	1.18	2998	1.16	2994	1.16	2996	1.16	3051	1.18	3099	1.18
150	10	3237	1.24	3212	1.24	3214	1.24	3251	1.24	3339	1.28	3304	1.26
100	20	3717	1.42	3698	1.42	3719	1.42	3810	1.44	3951	1.50	3781	1.44
~100	40	5095	1.94	5093	1.94	5165	1.96	5364	2.02	5631	2.14	5148	1.94

C. Heterogeneous loss scenario

In this more realistic scenario, we assumed that users are equally distributed in different environments; each environment having its own unique value of p_D. We also assumed that there are so many users that the item demand probability distribution is constant across all the environments. It can be seen from Table V that using the *SF* derived from the optimum, weighted average of the user response time (4970 slots), leads to suboptimal results with respect to each environment; moreover, the users that suffer most are the ones with a lower *p* as they see a dramatic rise in the response time with respect to their ideal carousel-FEC settings.

Although the resulting response times are suboptimal from the user's perspective, they still provide an improvement at network level, because there is a 4% capacity gain and a gain of about 200 slots compared to a carousel-FEC designed for the worst case i.e., $p = 40\%$. To reduce the penalty incurred by users in good conditions, the careful use of multiple channels has to be considered. From the user's perspective, the best case would be to have several channels, each with carousel-FEC settings specific to one environment. However, from the network's viewpoint this approach might be deemed resource inefficient. The trick is to find an optimal balance between resource consumption and user satisfaction (download time).

TABLE V. COMPARISON OF OPTIMAL $\{S_r, SF\}$ VALUES UNDER THE TWO LOSS SCENARIOS ; $Z_I = 10$, $Z_0 = 1$, $\Theta = 0$; CODEC TYPE = HYBRID150

Mean packet loss p (%)	Homogeneous		Non-homogeneous	
	S_{opt}	SF	S_{opt}	SF
10	3212	1.24	4874	1.90
20	3698	1.42	4874	1.90
40	5093	1.94	5163.2	1.90

V. CONCLUSIONS

We have studied the use of small and large codes within the context of the integrated carousel-FEC layer, first introduced in [4], considering two scenarios for packet loss distribution amongst users, homogeneous and heterogeneous.

In the homogeneous scenario, the hybrid scheme, which feeds an item into either the small or large codec depending on its size, outperforms the small and large scheme in terms of resource consumption and/or download times at packet loss rates of 10% and beyond, which are met for mobile satellite links. In the heterogeneous case, the use of carousel-FEC parameters corresponding to a minimum for the weighted average download time was shown to be punitive for users in good conditions. However, these mean parameters offered gains over a system designed for the worst-case reception conditions.

In the future, ways of reducing the penalty observed by good users in heterogeneous environments shall be investigated starting from the careful use of multiple channels. The current analytical work shall also be complemented by simulations.

REFERENCES

[1] M. Chipeta, M. Karaliopoulos, B.G. Evans, B. Garnier, and L. Roullet, "Designing the reliable transport layer for Satellite Digital Multimedia Broadcasting," in Proc. 14th IST Mobile and Wireless Communications Summit, June 2005.

[2] 3GPP TS 26.346, "Multimedia Broadcast/Multicast Service; Protocols and codecs," version 6.1.0, June 2005.

[3] DVB Document A081, "Transmission system for handheld terminals (DVB-H)," June 2004.

[4] M. Chipeta, M. Karaliopoulos, B.G. Evans, and R. Tafazolli, "On the use of packet-level FEC and data carousels for the delivery of broadcast/multicast services to mobile terminals," in Proc. 61st IEEE VTC May/June 2005.

[5] N.H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," ACM Wireless Networks, vol.5, no. 3, May 1999, pp. 171-182.

[6] M. Luby et al., "The use of FEC in reliable multicast," IETF RFC 3453, December 2002.

[7] V. Roca and C. Nuemann, "Design, Evaluation and Comparison of Four Large Block FEC Codecs, LDPC, LDGM, LDGM Staircase and LDGM Triangle, plus a Reed-Solomon Small Block FEC Codec," INRIA Research Report No. 5225, June 2004. Available: http://www.inrialpes.fr/planete/people/roca/mcl/mcl_docs.html

[8] J.W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in Proc. ACM SIGCOMM '98, pp. 56-67, September 1998.