

# SPARSE FEC CODES FOR FLEXIBLE MEDIA PROTECTION

*Jari Korhonen<sup>1</sup> and Pascal Frossard*

Ecole Polytechnique Fédérale de Lausanne (EPFL)  
Lausanne, 1015 - Switzerland

## ABSTRACT

In this paper, we study block codes that are optimized to recover some lost source data even in case when full recovery is not possible. Conventionally, block codes designed for packet erasure networks are aimed to recover all the lost source packets, assuming that the amount of lost data does not exceed the redundancy overhead. Unfortunately, this approach leads to poor performance if the fraction of lost data even occasionally exceeds the limit for full recovery capability. Recovery of part of the data may prove to be beneficial, especially when media data packets are unequal in importance. We present a short linear block code design that improves the performance of traditional Minimum Distance Separable (MDS) codes by reducing the fluctuation of the residual packet loss rate. These new codes also lead to a flexible design for unequal error protection of the media packets.

*Index Terms* — Forward error correction, video streaming, multimedia communications

## 1. INTRODUCTION

Recovery from packet losses is one of the fundamental technical problems in packet-switched networking. Traditionally, the problem can be solved by retransmitting the lost packets (Automatic Repeat reQuest, ARQ) or transmitting redundant data that can be used to replace lost packets (Forward Error Correction, FEC). FEC is generally preferred to ARQ in interactive applications with stringent latency requirements like video telephony or streaming with short delays.

According to the conventional wisdom, a good FEC code should be designed so that the redundancy overhead required to fully recover the missing source symbols is minimized. A block FEC encoder takes  $k$  source symbols as input and generates  $n$  FEC symbols as output ( $n > k$ ). In an ideal case, FEC decoder can regenerate the original  $k$  symbols when any  $k$  symbols out of  $n$  have been received. The ideal performance can be achieved with MDS codes [1]. However, in case there are more than  $n-k$  lost FEC

symbols, none of the lost source symbols can be recovered with MDS codes. If the loss rate varies around  $n-k$  symbols per block, the observed residual source symbol loss rate would vary enormously from block to block. This might be problematic in multimedia communications where applications are very sensitive to high quality variations, even if they can tolerate the loss of some of the (least important) packets.

In this paper, we propose a new design for short linear block codes, whose operating range is wider than of traditional MDS codes. Some of the lost source symbols may therefore be recovered even in the case where MDS codes would fail, which alleviates the clustering of residual losses in certain FEC blocks. It certainly represents an important advantage in applications such as video streaming, where advanced error concealment techniques can be effectively used to recover a small number of occasional packet losses, but not for heavily clustered packet losses. In addition, the flexible code design leads to efficient Unequal Error Protection (UEP) schemes that are well adapted to media communication problems. We analyze the performance of the sparse FEC codes, and experimental results show their advantages with respect to traditional codes.

## 2. BACKGROUND AND RELATED WORK

There are several different alternatives to implement FEC codes for different applications. Low Density Parity Check (LDPC) codes and their more advanced derivatives, such as Raptor codes, are often used for long blocks instead of MDS codes, due to the lower computational complexity compared to Reed-Solomon [2].

Traditionally, LDPC codes and their derivatives have been designed with intention to recover the random loss of a fraction  $p$  of the transmitted symbols at high probability, when the code rate  $R = 1 - p(1 + \epsilon)$  and  $\epsilon > 0$  (in an optimal case,  $\epsilon$  is as small as possible) [3]. A different approach has been taken recently with growth codes [4,5], intending to maximize the number of recovered symbols when  $p \geq 1 - R$ . The capability of recovering some data in case where traditional MDS or LDPC codes would fail may be useful

---

<sup>1</sup> Jari Korhonen is currently with Norwegian University of Science and Technology (NTNU), Trondheim, Norway

for applications, such as sensor networks in emergency scenarios [4] and UEP in video streaming [5].

The Partial Reed-Solomon (PRS) codes proposed in [6] target a similar objective. In PRS, source data is only partially protected by RS codes and the remaining part is left unprotected. With this method, the average recovery rate can be improved when the loss rate occasionally exceeds  $1-R$ , because the same FEC overhead is used to cover smaller amount of source data than with RS covering everything. The disadvantage, however, resides in the weak performance at low loss rates, as some of the unprotected packets may get lost and they cannot be recovered by the partial FEC codes.

### 3. SPARSE FEC CODE DESIGN PRINCIPLES

Due to latency and buffering restrictions in multimedia streaming applications, the application of FEC across media packets is generally restricted to short block codes. Systematic RS codes are certainly advisable in this context, due to their better performance compared to LDPC codes with short block lengths, when the complexity remains limited. However, it is even less probable to achieve perfect error recovery with short source blocks than long blocks.

An RS code can be defined as a system of linear equations, computed using Galois field arithmetics. A systematic FEC block is composed of a source data vector  $D=\{d_j, j=1..k\}$  and a FEC code vector  $C=\{c_i, i=1..n-k\}$ . Each FEC symbol can be computed from Eq. (1), where  $\alpha_{i,j}$  is a coding coefficient in a generator matrix  $G$ . If  $\alpha_{i,j}=j^{i-1}$ , the rows in  $G$  are orthogonal and the lost source symbols can be recovered from the system of equations, assuming that there are more than  $k$  symbols available.

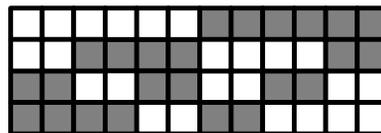
$$c_i = \sum_{j=1}^k \alpha_{i,j} d_j \quad (1)$$

Usually, FEC codes are not able to recover lost packets if less than  $k$  packets are received, while they recover all the lost packets in the contrary. However, it might be beneficial for some applications to trade-off perfect recovery for smoother recovery performance, and rather to augment the number of recovered symbols even when the actual loss rate is higher than expected. We propose a sparse RS generator matrix structure, where some of the coding coefficients  $\alpha_{i,j}$  are set to zero. For a comparable channel rate, such a code is not capable of perfect recovery of data in all the cases where conventional RS succeeds. However, partial recovery of data becomes possible with high probability even if (slightly) more than  $n-k$  packets are missing, so that the residual loss rate variations are reduced. The design of this kind of codes is partially inspired by the design of priority random linear codes used in network coding [7], and applied here to FEC protection of media streams.

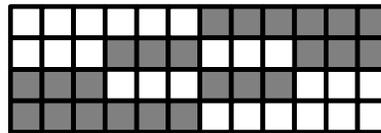
Heuristics are necessary for the design of sparse RS generator matrices. First of all, non-zero coefficients at each

row must partially overlap with non-zero coefficients at some other rows. Otherwise, the design would actually degrade to a system of independent RS codes with shorter source block lengths. Second, to achieve equal protection level for each source symbol, every row and column should have the same number of non-zero coefficients.

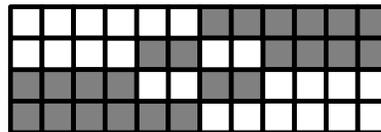
We show in Figure 1 a few examples of codes that satisfy these constraints, for a (16,12) code (i.e.,  $k=12$ ,  $n=16$ ), where the matrix  $G$  is represented with zero and non-zero coefficients in white and black squares, respectively. For such a code and the constraints above, only two levels of sparseness are feasible, which are 25% of zero coefficients (3 zeros per row, 1 zero per column) and 50% of zero coefficients (6 zeros per row, 2 zeros per column). With 25% of zeros, the 9 non-zero coefficients on each row must overlap with exactly 6 non-zero coefficients on every other row. With 50% of zeros, considering the requirements for partial overlap, there are several possible constructions of the sparse matrix  $G$ , two of them illustrated in Figure 1. The design varies with the sequence of overlapping non-zeros coefficients in different rows: the 6 non-zero coefficients in the first row can overlap with 2 non-zero coefficients on rows (2-2-2 overlap, Figure 1a), with 3 non-zero coefficients on two other rows and none on one other row (3-3-0 overlap, Figure 1b), or with four non-zero coefficients on one row, two on another and none on the last row (4-2-0 overlap, Figure 1c). With 75% of zero coefficients (9 zeros per row, 3 zero per column) the requirement for partially overlapping non-zeros on different rows cannot be achieved.



a) 2-2-2 overlap



b) 3-3-0 overlap



c) 4-2-0 overlap

Figure 1. Different possible designs for sparse (16,12) RS codes (50% zeros). White squares denote zero coefficients.

The linear system of equations corresponding to the matrix  $G$  can be solved if enough symbols are available at the decoder. Therefore, we can analyze the performance of such

codes with respect to the number of lost symbols. When the number of lost symbols is  $l$ , the average fraction of symbols  $R(l)$  that can be recovered is computed from Eq. (2), where  $C(i,l)$  is the proportional amount of combinations (out of all possible combinations) of  $l$  lost symbols in the  $n$ -symbol FEC block that can recover exactly  $i$  source symbols ( $i \leq l$ ).

$$R(l) = \frac{1}{l} \sum_{i=1}^{\min(l, n-k)} iC(i,l) \quad (2)$$

For conventional R-S codes, we have  $C(i,l)=1$  when  $l \leq n-k$ , and  $C(i,l)=0$  otherwise. With sparse R-S codes, the function  $C(i,l)$  depends on the number of zeros and the pattern of overlapping non-zero coefficients. In most cases, it cannot be written in a simple closed form. Figure 2 shows the values of  $R(l)$  derived by experimenting all possible combinations of packet losses. For the sake of clarity, the results are shown for only some of the possible overlap patterns.

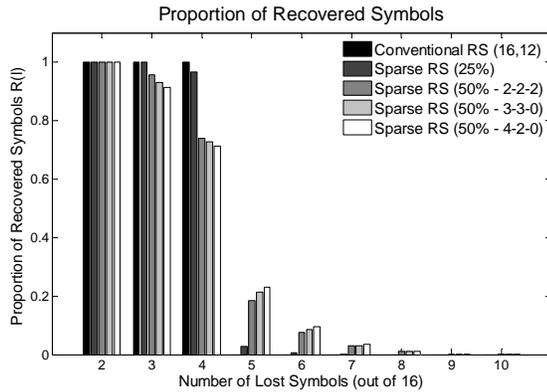


Figure 2. Amount of available symbols after decoding different FEC codes, respect to the number of lost symbols.

Assuming that the packet loss probability  $p$  is the same for every packet and that losses occur independently, the probability  $P(p,l)$  that exactly  $l$  packets are missing in a block of  $n$  packets is given by Eq. (3). By combining Eqs (3) and (2), it is possible to compute the average ratio  $R_{tot}(p)$  of recovered symbols as a function of  $p$ , as given in Eq. (4). The optimal FEC code can therefore be chosen among candidates that maximizes the error recovery rate. Figure 3 compares the measured and theoretical error correcting capabilities of conventional (16,12) RS codes and 50% sparse RS codes (with 3-3-0 overlap pattern). It shows that the experimental results follow closely the results derived analytically from Eq. (4).

$$P(p,l) = \binom{n}{l} p^l (1-p)^{n-l} \quad (3)$$

$$R_{tot}(p) = \frac{1}{n} \sum_{l=1}^n lP(p,l)R(l) \quad (4)$$

One important advantage of the flexible design of sparse codes resides in their ability to perform UEP for data with

different levels of priority. The degree of protection of each source symbol is related to the number of non-zero coefficients in the corresponding column of  $G$ . UEP can therefore be tuned finely to the relative importance of the data by adapting the design of the generator matrix. A practical example of such a matrix is shown in Figure 4, where we assume that the source symbols 1-4 belong to the highest priority class, symbols 5-8 to the intermediate and symbols 9-12 to the low priority class.

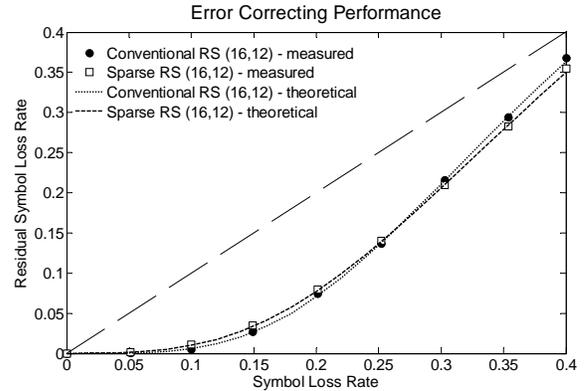


Figure 3. Residual loss rate ( $p-R_{tot}(p)$ ) of conventional and sparse RS codes illustrated.

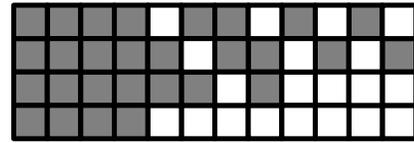


Figure 4. Binary representation of a generator matrix  $G$  optimized for UEP. White squares denote zero coefficients.

## 4. SIMULATIONS

In order to demonstrate the practical applicability of the proposed code, we have applied the residual packet loss patterns generated in simulations to real encoded H.264/AVC files to simulate a packet erasure channel. We have used the JM reference codec version 12.4 [8], ‘Soccer’ (CIF) sequence encoded at 1 Mbps. Flexible Macroblock Ordering (FMO) has been enabled to avoid aggressive quality degradation and fluctuation when the loss rate increases. The experiments have been repeated four times with different error patterns. The average PSNR values are shown in Figure 5. The results show that the proposed code improves the overall quality slightly at all loss rates larger than 0.1. The performance is even similar to the conventional RS codes at lower loss rates, since the sparse code decreases the harmful effect of clustered packet losses. This explains the small benefit of the sparse codes, even though the average residual packet loss rate at low loss rates is slightly lower for the conventional RS codes.

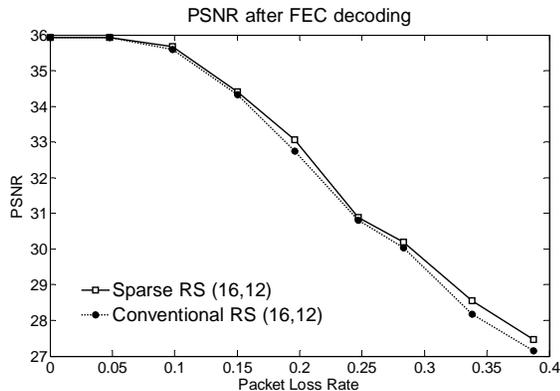


Figure 5. Video PSNR comparisons of sparse and conventional RS codes, as a function of the loss rate.

We study now the benefits of the proposed code for unequal error protection. We have also applied the sparse code shown in Figure 4 to a real H.264/AVC video stream and compared the performance to the conventional RS FEC and a simple conventional UEP scheme using RS (7,4) for high priority data, RS (5,4) for the intermediate priority and no protection for the low priority data. We have chosen a configuration with cycles of one I-frame, followed by four P-frames. This configuration permits a fair comparison to the baseline FEC with even error protection.

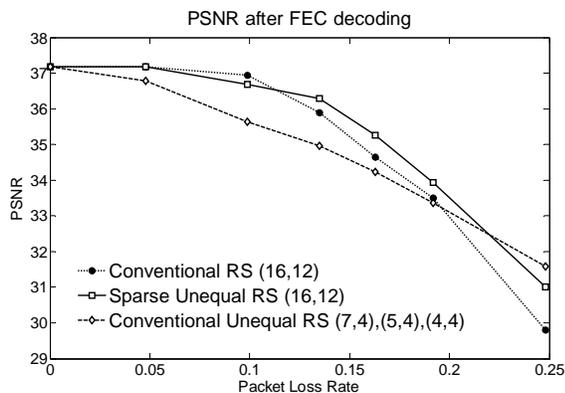


Figure 6. PSNR values for the proposed UEP FEC code, conventional RS and conventional UEP compared.

The highest priority class is assigned to I-frames (four packets per I-frame), the intermediate priority class to the two preceding P-frames and lowest priority level to the last two P-frames (two packets per each P-frame), resulting cycles of 12 NALUs, 4 in each priority class. The test bitstream is 'Foreman' (QCIF), coded at 256 kbit/sec bitrate with the FMO mode enabled. Figure 6 shows the PSNR quality compared to the one obtained with even protection and a conventional RS code. Preliminary results with different sequences show similar behavior, although not shown here due to the lack of space. The conventional RS

code performs slightly better at low loss rates, because the UEP scheme suffers from small number of low priority packet losses whereas equal FEC provides nearly perfect recovery of all packets. In contrast, the simple UEP becomes effective only at very high loss rates. The benefit of the proposed UEP is clearly shown at loss rates above 0.1. Even though the average residual packet loss rate is slightly higher with UEP than with conventional RS codes, the resulting quality is better. In addition, the quality fluctuation is reduced. The I-frames receive a better protection, which reduces the impact of error propagation.

## 5. CONCLUSIONS

In this paper, we have proposed sparse FEC codes that are optimized for partial recovery of data in the cases where conventional MDS FEC codes fail to recover any lost symbol. Such codes permit to reduce the quality variations, which is very important in media streaming applications. Their flexible design also leads to effective uneven error correction algorithms. The performance of the code has been validated with simulations of H.264/AVC video streaming. It has been shown that the proposed code results in improved quality due to unequal error protection of video packets. At the same time, it leads to smooth quality degradation when the packet loss rate increases, without significantly sacrificing on the quality at very low packet loss rates.

## 6. REFERENCES

- [1] S. Lin, and D.J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
- [2] A. Shokrollahi, "Raptor Codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551-2567, June 2006.
- [3] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," *Proc. of STOC '97*, El Paso, Texas, USA, pp. 150-159, May 1997.
- [4] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein, "Growth Codes: Maximizing Sensor Network Data Persistence," *Proc. of SIGCOMM '06*, Pisa, Italy, pp. 255-266, September 2006.
- [5] A. Dimakis, J. Wang, and K. Ramchandran, "Unequal Growth Codes: Intermediate Performance and Unequal Error Protection for Video Streaming," *Proc. of MMSP '07*, Chania, Greece, pp. 107-110, October 2007.
- [6] S. Karande, and H. Radha, "Partial Reed-Solomon Codes for Erasure Channels", *Proc. of ITW '03*, Paris, France, pp. 82-85, March 2003.
- [7] Y. Lin, B. Li, and B. Liang, "Differentiated Data Persistence with Priority Random Linear Codes," *Proc. of ICDS '07*, Toronto, Canada, June 2007.
- [8] H.264/AVC Reference Software Archive. Available online: [http://iphome.hhi.de/suehring/tml/download/old\\_jm/](http://iphome.hhi.de/suehring/tml/download/old_jm/)