A Proposed Fountain Unequal Erasure Protection Scheme for MPEG Transport Streams over IP

Claudio Alejandro Szabas
Weiler Alves Finamore

***Abstract*— The real-time transport of multi-media over IP faces issues, such as packet drops and jitter, that might generate severe impairments in the content being decoded at the reception. Channel coding is the most effective measure for overcoming these issues, since the latency imposed by retransmission protocols is not desirable. There are channel coding schemes specified in recommendations and standards, widely adopted by equipment vendors today. Among these, Fountain Codes present attractive characteristics for such applications.**

This article proposes an unequal protection scheme for Transport Streams over RTP/UDP/IP employing Fountain codes and presents comparative results of simulations performed with these and other channel coding schemes commonly adopted today.

***Keywords*— Channel coding, Fountain Codes, Reed-Solomon, MPEG Transport Streams, User Datagram Protocol, Internet Protocol.**

I. Introduction

There is a growing demand for the transport of video over IP today. Distribution of multi-media content over the Internet, contribution links inside traditional broadcasters' networks or standard and high definition contents transported in IPTV networks are a few examples.

IP networks, which can be modeled as *Packet Erasure Networks* (PEC), were not originally designed for the transport of real time multi-media. The main challenge is to overcome service-affecting issues, which result from latency, packet drops and jitter, while aiming to optimize the available network bandwidth.

The scenario presented in Fig. 1 illustrates a typical topology of an IPTV network. This scenario encompasses the most important concepts and evaluation mechanisms, employed in the simulations which will be discussed.

In general, such network includes three main locations types: a Main Hub, multiple Remote Head-ends and DSLAM facilities. The Main Hub receives contribution feeds, mostly over satellite, that will be re-multiplexed at Transport Stream level, encapsulated over IP and transmitted to Remote Head-Ends over a core network. In these locations, regional content is added and the end program is transmitted to DSLAM facilities over a limited capacity access network. Finally, the IPTV streams are multiplexed with other services such as broadband access and voice, for distribution to the end user, i.e., over DSL lines.

Centre for Studies in Telecommunications, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, E-mail: claudio.szabas@gmail.com, weiler@cetuc.puc-rio.br, (06/56413-1).

Quality impairments to the A/V services reaching the end user, commonly observed in such topology, comprise: the core network is usually high speed, high capacity and robust, but the access network connected to it has lower capacity and might become overloaded with voice and broadband traffic and discard packets carrying IPTV content. Jitter might also be present in this hop. Furthermore, the last mile DSL circuit, feeding the end customer set-top-box, suffers more often from degradation, which also generates packet drops.

In order to quickly detect such problems, as indicated in Fig. 1, particular points in the network can be continuously monitored, as indicated in the figure. The set of measurements that can be employed include Transport Stream Measurements according to [5] and objective video quality measurements such as presented in [14].

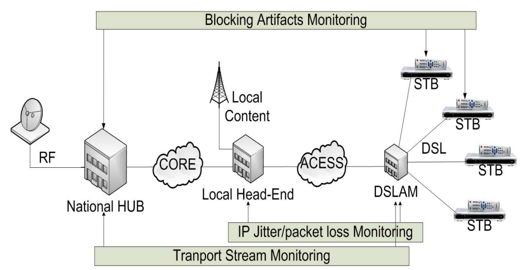

Fig. 1. Typical IPTV architecture

Some standards defined recently aim protection of professional video over IP. The RFC given in [11] specifies an RTP payload format for the transport of data protected with erasure protection codes. This RFC does not define channel coding parameters, but it cites Reed-Solomon and Hamming as optional schemes.

The Code of Practice [3] is widely adopted by equipment vendors today. It makes use of the payload format specified in [11] and moreover, it specifies how the input string has to be arranged with respect to dimension and interleaving of the input block. It also defines an optional second dimension for the channel coding scheme in order to cover a wider variety of erasure patterns. The first interleaved dimension is meant to cope with bursts of packet erasures while the second dimension to cope with single packet erasures, which might occur in addition to bursts.

Fountain Codes are sparse graph codes, with properties that make it attractive for transport of multimedia over IP. The main property associated to such schemes is that these have finite dimension and infinite block length and thus a rateless scheme can be implemented, for scenarios where the channel conditions are unknown to the sender prior to starting the communication. The overhead requirements are also low in Fountain coding schemes.

II. Simulation Scenarios

The simulations disclosed herein were repeated for the distinct erasure patterns exposed in [15], namely, random single packet erasures, random burst erasures and a combination of both.

A. Fountain Encoder

The Fountain encoder construction employed is shown in Fig. 2.

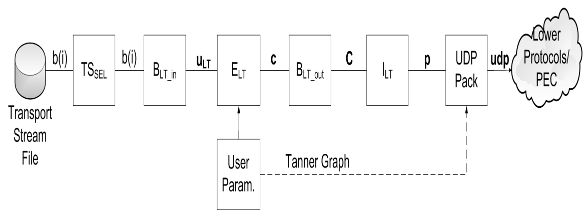

Fig. 2. LT Encoder

Where the following notation is used:

- $\mathbf{b}$(i) is the i-th individual byte read from the Transport Stream file.
- $\mathbf{TS}_{sel}$ is the sync evaluation block of each TSP.
- $\mathbf{B}_{LTin}$ is the LT encoder input buffer.
- $\mathbf{u}_{LT}$ is the sequence of input blocks for the LT encoder.
- $\mathbf{E}_{LT}$ is the LT encoder.
- $\mathbf{c}$ is the sequence of encoded blocks from the LT encoder.
- $\mathbf{B}_{LTout}$ is the encoder output buffer.
- $\mathbf{I}_{LT}$ is the LT interleaver.
- $\mathbf{p}$ is the sequence of UDP payload blocks.
- $\mathbf{UDP}$ is the UDP packetizer.
- $\mathbf{udp}$ is the sequence of network packets.

The $\mathbf{TS}_{sel}$ block analyzes the sync bytes of the incoming Transport Stream sequence and assures the selection of valid TSP's, which are fed to the input buffer $\mathbf{B}_{LTin}$.

In the LT encoder block, $\mathbf{E}_{LT}$, the process is achieved by simply multiplying every component vector $\mathbf{u}_{\mathrm{LT}}(i)$ belonging to the sequence of vectors $\mathbf{u}_{\mathrm{LT}}$ by the matrix $\mathbf{G}$, of dimensions (k, n), which characterizes the LT-encoder. The encoder output is a sequence of vectors of length n, given by $\mathbf{c} = \big(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_\beta\big)$, where β is the amount of encoded blocks produced within the interval examined and each component vector is given by the notation:

$$\mathbf{c}_i = \big(c_i(1), c_i(2), \ldots, c_i(n)\big). \tag{1}$$

The *User Parameters* provided to $\mathbf{E_{LT}}$ are given in [9] and define the degree distribution employed in the construction of the Generator matrix. We make use of the *Soliton* degrees distribution presented in this same reference.

In the output buffer, each vector $\mathbf{c}(i)$ of the sequence $\mathbf{c}$, is stored as a column of the matrix $\mathbf{C}$, of dimension $[n, \beta]$, where n is the length of the LT encoded block and β the amount of encoded blocks generated within the time interval observed. We chose a value of β, such that an integer amount of lines is picked in order to compose one network packet. The required output buffer capacity is then given by $n \cdot \beta$.

The interleaver provides the sequence $\mathbf{p}_i = \big(p_1, p_2, \ldots, p_\zeta\big)$, where ζ is the amount of $\mathbf{p_i}$ blocks generated within the interval observed.

Each component $\mathbf{p}_i$ is given by the concatenation of ν lines of $\mathbf{C}$:

$$\mathbf{p}_i = \big(C(i,:), C(i+1,:), \ldots, C(:, i+(\nu-1)\big) \tag{2}$$

Hence, each vector $\mathbf{p}_i$ has size $\nu \cdot \beta$ bytes, being n the LT encoder output block size. As a result, the interleaving operation can be characterized as follows:

$$\mathbf{p}_i = \big(c_1(i), c_2(i), \ldots, c_\beta(i), \ldots, c_\beta(i+(\nu-1)\big) \tag{3}$$

where a zig-zag scan is accomplished across ν lines of the matrix $\mathbf{C}$. In other words, each network packet $\mathbf{udp_i}$ will contain ν lines of $\mathbf{C}$.

For the sake of simplicity, in the simulation herein, β is taken to be the size of a TSP. Thus, considering the IP Maximum Transfer Unit of $1,500$ bytes, ν equals seven.

The user parameters employed in the LT encoder have to be provided to the LT decoder. We assume that the same is appended unscrambled to the first bytes of the payload section of each UDP packet. This is, for example, the method suggested in [4].

At the decoding side, lower level mechanisms, such as packet identification provided by the *Real-Time Transport Protocol*, can inform the LT decoder about the missing packets, which will be handled as "erasures" by the LT decoder.

B. Reed-Solomon

The single- and two-dimensional Reed-Solomon schemes employed herein reproduce the framework defined in the Pro-MPEG FEC Code of Practice [3].

The single dimensional RS encoder has the same signal flow as the LT encoder exposed in the last section, except for the $\mathbf{E}_{LT}$ which is replaced by an RS encoder block $\mathbf{E}_{LT}$.

Fig. 3 shows the block diagram of the two-dimensional Reed-Solomon encoder implemented.

For notational purposes:

- $\mathbf{I}_{RS}$ is the RS interleaver for the first dimension only.
- $\mathbf{E}_{RS}(i)$ is the RS encoder for the each dimension.
- $\mathbf{B}_{out}(i)$ is the output block packing buffer for each dimension.

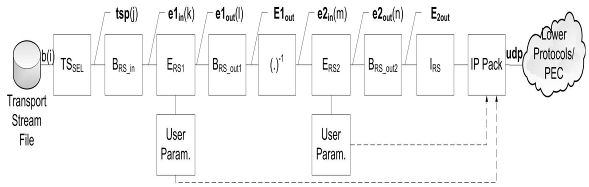

Fig. 3. RS-2D Encoder

C(i) is the matrix composed by the output block sequence of each dimension. $C(1)$ and $C(2)$ have sizes $[k_{RS1}, N_{RS1}]$ and $[k_{RS2}, N_{RS2}]$, being k_{RS1} and k_{RS2} the source block sizes and N_{RS1} and N_{RS2} output block sizes for the RS encoders of the first and second dimensions, respectively.

RS is the packetizer responsible for merging the original payload and both overheads of the two separate dimensions into the same matrix.

S is the matrix of size $[N_{RS1}, N_{RS2}]$, which contains both overheads from the two dimensions and the original payload bytes.

The overhead blocks generated by the second dimension of the code are indicated by *FEC'*. According to [3], the second dimension is intended to cope with single packet losses that might happen in addition to burst erasures.

The matrices C_1 and C_2, which result from storing the encoded blocks as its rows, are typically rectangular. Three resulting structures, namely, payload, overhead for first dimension and overhead for second dimension, are arranged in such a way that each UDP packet, the atomic unit for packet erasures, will not contain bytes belonging to different structures, as given in [11].

C. Results

The degradation of the decoded content when the code's rate is modified was observed. The measured degradation in this situation reflects the probability of unrecovered symbols in the decoded content.

The code's rate is varied within a range that allows us to observe the transition from a very good quality decoded content, down to a very impaired decoded content, while the channel erasure probability P_{er} is kept constant at 0.03.

1) Random burst erasures: Fig. 4 shows the performance results for both Reed-Solomon schemes and the LT coding scheme. The Reed-Solomon codes present a much better performance until a value of $N \approx 1.08 \cdot k$, at which point, any new LT encoded symbol that arrives successfully at the LT decoder, contributes significantly with the LT decoding process and the same outperforms the Reed-Solomon schemes.

Another interesting aspect, is that the second dimension of Reed-Solomon does not provide improvement for the *burst-only* erasure pattern. Actually, it increases the overhead at no significant benefit, decreasing performance measured against overhead cost, making the single dimensional Reed-Solomon advantageous for this type of erasure pattern.

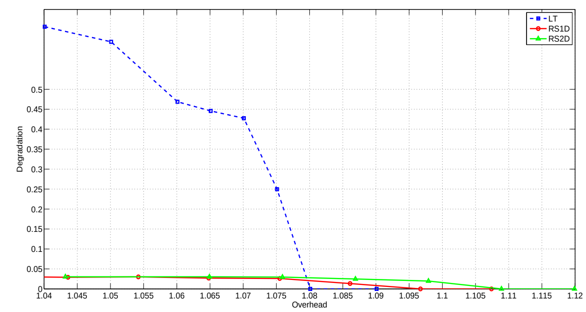

Fig. 4. Performance Comparison for random burst packet erasure

2) Random single packet erasures: In this case, the two dimensional and the single-dimensional Reed-Solomon schemes present very proximal performance, as shown in Fig. 5. No significant performance variation is observed for the LT scheme.

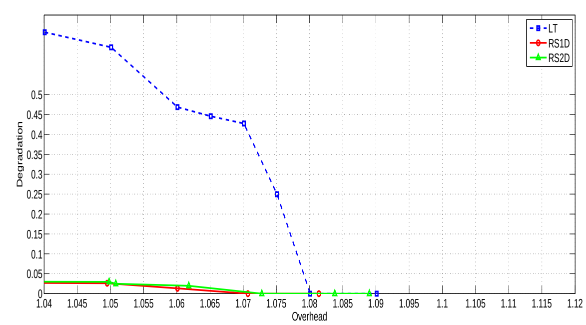

Fig. 5. Performance Comparison for random single packet erasures

3) Combination of Random single packet erasures and random burst erasures: This erasure pattern is said to be more realistic according to [15]. In the channel simulation herein, approximately half of the erasures are distributed as single packet erasures, whereas the other half as bursts of 7 packets being erased.

It can be noted that in this case the second dimension for the Reed-Solomon scheme provides an improvement over the single-dimensional one. The LT scheme still outperforms the Reed-Solomon schemes at n approx. 1.08.

4) Visual impairments: Figures 7, 8 and 9 show snapshots of samples resulting from the LT, the single- and the two-dimensional Reed-Solomon processes respectively, at the point of 1.08 overhead shown in fig. 6.

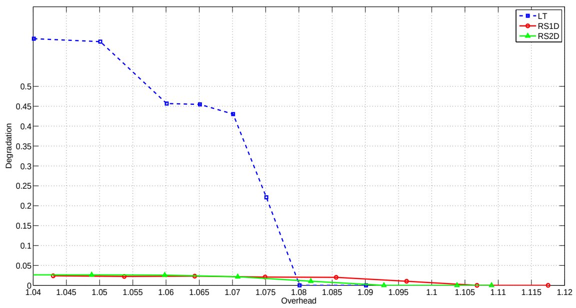

Fig. 6. Performance Comparison for random burst and single packet erasures

Fig. 7. Visual impairments for LT at $N \approx 0.8k$

Fig. 8. Visual impairments for RS1D at $N \approx 0.8k$

Fig. 9. Visual impairments for RS2D at $N \approx 0.8k$

III. Unequal Fountain Scheme

It was noted that, for some decoded samples, the byte degradation obtained was within satisfactory levels, but time-outs were observed during presentation. Moreover, it was observed that these cases presented PSI and/or PCR corruption. In order to avoid such cases, we propose an unequal protection scheme for protection of Transport Streams over IP networks, that privileges packets carrying PSI and PCR information.

The proposed workflow is shown in figure 10. At the encoder, when the Transport Stream file is read, a parallel block is analyzing the original TSPs. This is not intended to be a complete analysis, only of the fields of interest, at low processing costs.

For the selection of PCR TSP's, the items analyzed include the Adaptation Field and the PCR presence flags which are explained in [1]. If both bits are set, the *TS Analysis block* will switch the incoming TSP to the proper path employing a more robust version of the Generator matrix.

For PSI information, the "TSP Analyzer" block first looks for the known PAT PID, where it finds the updated list of current PMT's and from this point on, the same are added to the search criteria.

Since it was observed, in the LT code performance presented in Fig. 10, that, when close to decoding completion, an increment between one and two percent in the overhead is capable of significantly improving the decode-ability of up to the entire block, we made the choice of providing an additional four percent of LT encoded symbols over the current code's rate, upon detection of PCR or PSI, i.e., twice the overhead increment capable of assuring successful decoding for the present channel.

Considering that PSI and PCR information may represent approximately one or two percent of the total data which is transported by the stream, this method requires very little additional overhead in order to assure continuous playback and overall improved presentation at the reception.

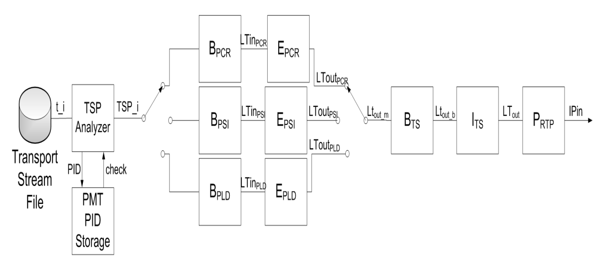

Fig. 10. Adaptive LT encoder considering PSI and PCR

- $\mathbf{t}(i)$ is the i-th bit entering the encoder, resulting from the reading process.
- $\mathbf{B}_{TS}$ is the Transport Stream packet buffer, of the size of one TSP of 188 Bytes.
- $\mathbf{tsp}(j)$ is the j-th integral Transport Stream packet stored in B_{TSP} being fed to the analysis block.
- $\mathbf{TS}_{Anal}$ this block will be checking for presence of PSI or PCR information in the TSP.

PMT This block will store the PMT pid for the single program – in this section we evaluate an SPTS – once the same is obtained from the PAT. It will be used for checking PMT presence in the upcoming packets.

$\mathbf{B}_{TS}$ is the Buffer for the string composed of LT_{PCR}, LT_{PSI} and LT_{PLD}.

$\mathbf{I}_{TS}$ Interleaver for the string composed of LT_{PCR}, LT_{PSI} and LT_{PLD}.

$\mathbf{P}_{RTP}$ Packetizer of the adaptive encoding scheme into RTP/UDP/IP packets.

We tested this scheme with overhead values of 1.075 and 1.085, at which point, very small additional overhead of approximately $1-2$ percent will suffice to successfully decode the original input block. Hence, we increment the overhead for PSI and PCR TSP's by 4 percent, up to 1.12, what assures decoding success for these packets at insignificant the cost of less than one percent of the total overhead. As a result, no PCR or PSI errors were found in the decoded string and continuous presentation was assured.

IV. Conclusions

Considering the performance curve of figure 6, we conclude that Reed-Solomon is the scheme of choice for bandwidth critical applications, since at lower overheads — or higher loss rates — the LT decoder recovers almost no information, while the Reed-Solomon decoder is capable of recovering a significant amount of the original symbols. However, if the application is quality critical, in other words, if it is worth to have much higher quality, i.e., the original content completely decoded, at the cost of a slightly larger overhead, the LT code is the appropriate choice for this simulation scenario.

The performance for the LT code was more or less the same for all erasure patterns employed. When the erasure pattern comprises random bursts of packets only, the second dimension of the Reed-Solomon scheme does not present any improvement over the single dimensional one, instead, it makes the performance worst. This probably results from the fact that the second dimension does not benefit from the interleaving, which distributes the burst losses and hence, it is not recovering a significant amount of erased symbols, that justifies the increase in overhead that it imposes.

When the combined erased pattern is employed, adding random single packet erasures to the random burst pattern, the second dimension copes significantly with the decoding process, whereas the first interleaved dimension is more severely impaired and presents the worst performance. When single packets are erased at random, with no burst erasures, both Reed-Solomon schemes present very similar results.

Finally, the proposed unequal protection scheme providing privileged protection to relevant Transport Stream parameters eliminates the presentation time-outs still observed with the "Flat LT" scheme at low byte degradation rates, at the cost of very small additional overhead, less than one percent of the encoded symbols' string for the erasure rates employed in the simulations.

Acknowledgement

To my colleagues at Miranda Technologies, TV Globo and Globosat.

Referências

[1] ISO/IEC 13818-1, *Information Technology — Generic Coding of Mooving pictures and associated audio information: Systems*, Second Edition, 2000-12-01.

[2] European Broadcast Union, *A Proposed Time Stamped Delay Factor algorithm for measuring Network Jitter on RTP Streams*. EBU TECH 337, 2010.

[3] Pro-MPEG Code of Practice 3 release 2, *Transmission of Professional MPEG-2 Transport Streams over IP Networks*, July 2004.

[4] DVB Document A086, *Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP*, July 2004.

[5] European Broadcasting Union, ETSI ETR 101 290 V1.2.1 (2001-05), *Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems*, 2001.

[6] J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[7] A. Shokrollahi. *Raptor Codes*, Digital Fountain, Inc., École Polytechnique Fédérale de Lausanne, January 13, 2004.

[8] A. Shokrollahi. *Fountain Codes*, École Polytechnique Fédérale de Lausanne.

[9] M. Luby. *LT Codes*, Digital Fountain, Inc., 2002.

[10] M. Luby, M. Watson, L. Vicisano, *Forward Error Correction (FEC) Building Block*, RFC5052, Network Working Group, August 2007.

[11] J. Rosenberg, H. Schulzrinne, *RFC2733 - An RTP Payload Format for Generic Forward Error Correction*. Columbia University, 1999.

[12] X. Zhao, J. Guo, C.T. Chou, S.K. Jha, *Resilient Multicasting in Wireless Mesh Networks*, School of Computer Science and Engineering of the University of New South Wales, 2004.

[13] D. Hoffman, G. Fernando, *RTP Payload Format for MPEG1/MPEG2 Video*, RFC 2250, January 1998.

[14] R. Muijs, I. Kirenko, *A No-Reference Blocking Artifact MEasure for Adaptive Video Processing*, Philips Research Laboratories.

[15] Van der Schar, Michaela, Chou, Philip, *Multimedia over IP and Wireless Networks, Compression, Networking and System*, Elsevier Inc., 2007.