EXPANDING WINDOW FOUNTAIN CODES FOR SCALABLE VIDEO MULTICAST

Dejan Vukobratović †, *Vladimir Stanković* ‡, *Dino Sejdinović* +, *Lina Stanković* ‡, *Zixiang Xiong* *

† Dept. of Communications and Signal Processing, University of Novi Sad, Novi Sad, Serbia.
‡ Dept. of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK.
+ Dept. of Electrical and Electronic Engineering, University of Bristol, Bristol, UK.
* Dept. of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA.

ABSTRACT

Digital Fountain (DF) codes have recently been suggested as an efficient forward error correction (FEC) solution for video multicast to heterogeneous receiver classes over lossy packet networks. However, to adapt DF codes to low-delay constraints and varying importance of scalable multimedia content, unequal error protection (UEP) DF schemes are needed. Thus, in this paper, Expanding Window Fountain (EWF) codes are proposed as a FEC solution for scalable video multicast. We demonstrate that the design flexibility and UEP performance make EWF codes ideally suited for this scenario, i.e., EWF codes offer a number of design parameters to be "tuned" at the server side to meet the different reception conditions of heterogeneous receivers. Performance analysis of H.264 Scalable Video Coding (SVC) multicast to heterogeneous receiver classes confirms the flexibility and efficiency of the proposed EWF-based FEC solution.

***Index Terms*—** Digital Fountain Codes, H.264 SVC, Scalable Video Multicast, Unequal Error Protection

1. INTRODUCTION

Efficient multicast transmission of scalable video content over lossy packet networks to heterogeneous receivers is still a challenge. Scalable video coding techniques enable the receivers to progressively improve their reconstructed video quality with the amount of the data received. This may enable receivers with increased capabilities, such as available bandwidth and screen resolution, to experience better video quality, while at the same time providing the basic reconstruction quality for low capability receivers.

However, even for high capability receivers, packet losses in scalable video content transmission can significantly deteriorate the quality of the reconstructed data. For example, an early packet loss in the transmission of a typical scalable coded data segment, where the data importance decreases along the data segment, may lead to severe error propagation. For this reason, scalable video is usually protected at the server side using forward error correction (FEC) mechanisms before being multicast.

Maximum distance separable (MDS) codes, such as Reed-Solomon (RS) codes, have been the traditional FEC solution for real-time multimedia delivery. However, recently, Digital Fountain (DF) codes, such as LT [1] or Raptor codes [2], have proved to be a more flexible and efficient FEC solution for multicasting scalable video over lossy packet networks [3][4]. DF codes can provide linear encoding/decoding complexity and universal capacity-approaching behavior for any channel packet loss probabilities for the price of a small reception overhead, as compared to RS codes. However, two major drawbacks of standard DF solutions for scalable video multicast applications are identified, namely: (i) standard DF codes are equal error protection (EEP) codes, whereas scalable video transmission calls for unequal error protection (UEP) FEC schemes due to the unequal importance of data in the scalable bitstream, (ii) if a minimum amount of DF encoded data is not received, the DF decoder can only reconstruct a small portion of the transmitted video block.

In this paper, a DF solution based on UEP DF codes, named Expanding Window Fountain (EWF) codes [5], is proposed that addresses both of the aforementioned problems. We apply the EWF codes [5] to H.264 SVC scalable coded video streaming. System setup for scalable video multicast to heterogeneous receiver classes is described in Section 2, followed by a review of EWF codes and their design parameters in Section 3. Section 4 provides an example of the EWF code optimization, where EWF codes are adapted to meet the different reception conditions of different receiver classes. In Section 5, the performance analysis of H.264 SVC coded multicast video streaming, optimized with respect to the end-to-end distortion performance, confirms the flexibility and efficiency of the proposed EWF FEC solution. The paper is concluded in Section 6.

2. SYSTEM SETTING

We consider the scenario where real-time scalable coded video stream is transmitted from a video server to a number of heterogenous receivers over a lossy packet network such as the Internet. At the video server side, the scalable coded

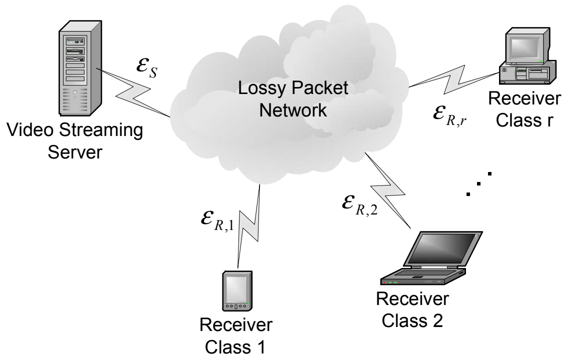

Fig. 1. Scalable video multicast to heterogeneous receivers.

video stream is periodically broken into source blocks, and each source block is separately encoded by a DF encoder. We assume that each source block consists of an equal number of K symbols, and that the importance of data decreases from the beginning towards the end of the block. Due to real-time constraints, the video server is able to produce "only" a finite amount of $\epsilon_S K$ encoded DF symbols before moving on to the next source block. The source overhead, $\epsilon_S > 1$, is determined by the video server capabilities and/or the bandwidth of the video server access link.

Encoded DF symbols are transmitted, in a multicast session, to heterogeneous receivers. We classify receivers into r receiver classes based on their capabilities and channel quality. The i-th receiver class, $1 \leq i \leq r$, is quantified using the reception overhead $\epsilon_{R,i}$, where $\epsilon_{R,i} \leq \epsilon_S$, i.e., the receiver in the i-th class is able to collect $\epsilon_{R,i}K$ encoded DF symbols of each source block, out of the $\epsilon_S K$ symbols transmitted. Additionally, we assume that $\epsilon_{R,i} < \epsilon_{R,j}$ if $i < j$, i.e., the receiver capabilities increase with the receiver class index i. This scalable video multicast setting is illustrated in Figure 1.

Our system aims at setting up a DF solution that adapts the real-time scalable video stream delay constraints and unequal data importance to different receiver classes. This scenario calls for the DF solution that possesses UEP and unequal recovery time (URT) properties. In other words, the more important part of the source block should be better protected and recoverable with as small reception overhead as possible. Our solution is based on the UEP DF codes named Expanding Window Fountain (EWF) codes [5], that we briefly review in the next subsection. We note that a related work, but using different approach based on the UEP DF codes introduced in [6], was recently proposed by Dimakis *et al.* [7].

3. EXPANDING WINDOW FOUNTAIN (EWF) CODES

EWF codes are a novel class of UEP DF codes based on the idea of "windowing" the source block to be transmitted. The sequence of expanding windows (subsets of the source block), where each window is contained in the next window in the sequence, is defined over the source block (Figure 2). The number of expanding windows applied is equal to the number of importance classes of the source block. The first and the most important symbol class is defined by the "innermost" window, and is protected by all the other windows in the sequence. The i-th importance class is the set of all input symbols that belong to the i-th window, excluding the symbols that belong to the $(i-1)$ window. The last and least important window contains all the symbols in the source block. The EWF code design generalizes the standard LT code design: LT codes are EWF codes defined by a single window, i.e., all the input symbols are of equal importance.

The set of expanding windows is characterized by a window selection probability distribution. The importance of the receiver class will dictate the selection probability of window. Upon window selection, a new EWF encoded symbol is generated by a suitably chosen degree distribution as if encoding were performed by a standard LT code only on the input symbols from the selected window. This procedure is repeated at the EWF encoder for each EWF encoded symbol.

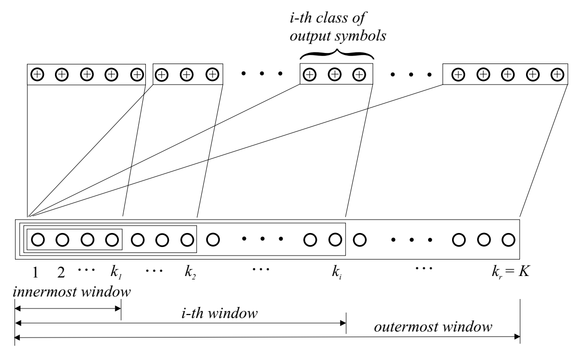

Fig. 2. Expanding Window Fountain Codes.

The EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ is defined using the set of polynomials $\Pi(x), \Gamma(x), \Omega^{(1)}(x), \ldots, \Omega^{(r)}(x)$. $\Pi(x) = \sum_{i=1}^{r} \Pi_i x^i$, where $\Pi_1 = \frac{k_1}{K}$ and $\Pi_i = \frac{k_i - k_{i-1}}{K}, 2 \leq i \leq r$, describes the division of the source block into the set of r expanding windows of size $k_i > 0$, where $k_i < k_j$ if $i < j$, and $k_r = K$. The selection probability distribution associated with the set of expanding windows is described using polynomial $\Gamma(x) = \sum_{i=1}^{r} \Gamma_i x^i$, where Γ_i is the probability of selecting the i-th window. The degree distribution $\Omega^{(j)}(x) = \sum_{i=1}^{k_j} \Omega_i^{(j)} x^i$ describes the LT encoding performed on the j-th window. To summarize, EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ assigns each encoded symbol to the j-th window of size k_j with probability Γ_j and encodes the data from the selected window using the LT code with the degree distribution $\Omega^{(j)}(x) = \sum_{i=1}^{k_j} \Omega_i^{(j)} x^i$.

4. EWF CODE OPTIMIZATION

In this section, we determine the EWF parameters tradeoff to optimize the reconstructed video quality. As a performance measure, we use the probability that the i-th importance class (which can be seen as the i-th quality layer) is completely recovered by the receiver in the j-th class. For a given EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$, the asymptotic erasure probability $y_{l,i}(\epsilon)$ (as $K \to \infty$) of the source symbols from the i-th importance class, after l iterations of the standard belief-propagation iterative decoder with the reception overhead ϵ, is given by recursion ([5], *Lemma 3.2*):

$$y_{0,j} = 1 \qquad (1)$$
$$y_{l,j} = e^{\left(-(1+\varepsilon)\sum_{i=j}^{r}\frac{\Gamma_i}{\sum_{t=1}^{i}\Pi_t}\Omega'^{(i)}\left(1-\frac{\sum_{m=1}^{i}\Pi_m y_{l-1,m}}{\sum_{t=1}^{i}\Pi_t}\right)\right)}.$$

We use these probabilities to approximate the erasure probability behavior in the finite source block length scenario. With this approximation, the probability $P_i^{(j)}$ that the i-th importance class is completely recovered by the receiver in the j-th class, after l iterations of the iterative decoding, is equal to $P_i^{(j)} = (1 - y_{l,i}(\epsilon_{R,j}))^{s_i}$, where s_i is the number of symbols in the i-th importance class ($s_i = \Pi_i K$, for $1 \leq i \leq r$).

The probabilities $P_i^{(j)}$ are useful to set up our EWF code design problem. Before doing so, note that $P_i^{(j)} < P_i^{(k)}$ for $j < k$, due to the fact that $\epsilon_{R,j} < \epsilon_{R,k}$ and *Lemma 3.2* [5]. In other words, if we set a desired performance threshold on the probability $P_i^{(j)}$ for the receiver class j, all the receiver classes $k > j$ will satisfy the same constraint. Therefore, it is convenient to place the performance constraints only on the probabilities $\boldsymbol{P_{th}} = (P_1^{(1)}, P_2^{(2)}, \ldots, P_r^{(r)})$. For a given reception overheads $\boldsymbol{\epsilon_R} = (\epsilon_{R,1}, \epsilon_{R,2}, \ldots, \epsilon_{R,r})$, the EWF code design problem is: Find the set of the EWF code design parameters $(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ such that the corresponding EWF codes satisfy the performance threshold $\boldsymbol{P_{th}}$ for the different receiver classes, given their reception capabilities $\boldsymbol{\epsilon_R}$. More elaborate exploration on the EWF code design scenarios for scalable data multicast can be found in [8].

The above design problem is illustrated by example. We assume a simple scenario with only two receiver classes, and two importance classes of the scalable video stream (the base layer and one enhancement layer). Thus, we need to find the set of the EWF parameters $(\Pi_1 x + (1 - \Pi_1)x^2, \Gamma_1 x + (1 - \Gamma_1)x^2, \Omega^{(1)}, \Omega^{(2)})$ such that the corresponding EWF codes provide a reconstruction probability not smaller than $P_1^{(1)}$ for the base layer for both of the receiver classes, and a reconstruction probability of at least $P_2^{(2)}$ for the enhancement layer for the receiver class with better reception conditions.

We assume that the source block length is equal to $K = 3800$ DF symbols. The degree distribution $\Omega^{(2)}(x)$ applied on the larger window (the whole source block) is the constant average degree Raptor distribution [2] (see also [5]). For the first window of higher importance, we use the "stronger" truncated robust solution distribution $\Omega_{rs}(K_{rs}, \delta, c)$ [5], obtained by limiting the maximum degree of the robust solution distribution [1] to K_{rs} irrespectively of the window size. In our example, we apply $\Omega_{rs}(250, 0.5, 0.03)$ distribution on the first window. The threshold probabilities and the reception capabilities are set to values $\boldsymbol{P_{th}} = (0.9, 0.5)$ and $\boldsymbol{\epsilon_R} = (0.35, 3.5)$, respectively. In other words, we assume two very disparate receiver classes, and our aim to provide the performance guarantees for the base layer for both classes, and for the enhancement layer only for the better class. We fix the degree distributions $\Omega^{(1)}$ and $\Omega^{(2)}$ in advance, which reduces our task to find the solution set of polynomials $\Pi(x)$ and $\Gamma(x)$ that satisfies the given constraints. For our two-window case, the solution is the set of all pairs (Π_1, Γ_1), where Π_1 describes the size of the first (more important) window, and Γ_1 gives the probability of selection of this window. The set of pairs (Π_1, Γ_1) that satisfy the given conditions is illustrated in Figure 3, within the darkest shaded region. This region is the intersection of regions corresponding to the performance conditions given by $P_1^{(1)}$, for the worse receiver class, and $P_2^{(2)}$ for the better receiver class.

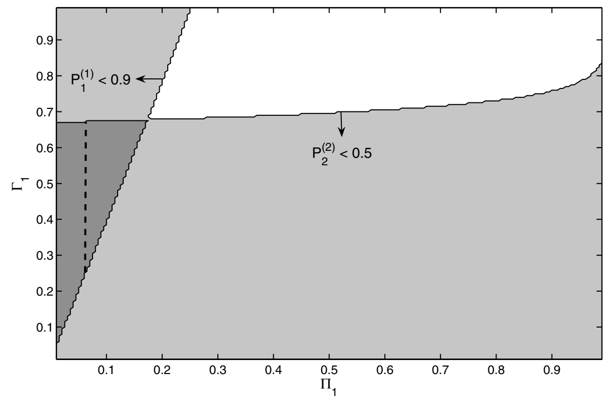

Fig. 3. EWF Code Optimization Example.

5. H.264 SVC VIDEO STREAMING ANALYSIS

In this section, we apply our design problem to a video server multicasting H.264 SVC stream. We assume the transmission of the CIF *Stefan* video sequence (30 fps, 352×288) with the base layer and four enhancement layers which gradually improve the overall video quality (Table 1). The sequence is segmented into GOFs of size 16 frames, and every $16/30$ seconds the EWF encoder is supplied by a new GOF data as the source block. The source block size is approximately 190000 bytes, and assuming DF symbol size of 50 bytes, we obtain the source block size of $K = 3800$ DF symbols (as analyzed in Section 4). The base layer is placed in the first of two EWF windows, with the first window size necessary to accommodate the base layer data set to $k_1 = 250$ DF symbols. All the enhancement layers, together with the first window, form the second window, i.e., the whole source block. With the same recovery probability constraints $\boldsymbol{P_{th}}$ and reception conditions $\boldsymbol{\epsilon_R}$ as in the previous section[1], the solution set is obtained as the dashed-line in Figure 3 (note that

[1]Note that the reception conditions $\boldsymbol{\epsilon_R} = (0.35, 3.5)$ corresponds to the receiver classes bitrates of approximately $\boldsymbol{\epsilon_R} = (1Mbps, 10Mbps)$.

Table 1. H.264 SVC Compressed *Stefan* Sequence.

Layer	Bit Rate [kbps]	Y-PSNR [dB]
Base Layer	183.884	32.53
1^{st} Enhancement Layer	366.364	33.63
2^{nd} Enhancement Layer	528.400	34.70
3^{rd} Enhancement Layer	1170.432	38.15
4^{th} Enhancement Layer	2845.011	41.40

$\Pi_1 = 250/3800 = 0.065$). In other words, it can be seen from Figure 3 that the interval $\Gamma_1 \in [0.25, 0.68]$ of the first window selection probabilities is the solution set for $\boldsymbol{P_{th}}$ and $\boldsymbol{\epsilon_R}$, given the predefined Π_1, $\Omega^{(1)}$ and $\Omega^{(2)}$.

In the following, we discuss the possibility of optimizing Γ_1 value for the optimal end-to-end video distortion performance. However, as it is clear from Figure 3, instead of optimizing Γ_1 for fixed Π_1, it is possible to place more data into the first window, i.e., to increase Π_1 to the maximum value $\Pi_1^{max} = 0.175$ that satisfies the same constraints. The Π_1^{max} value admits the whole first and large part of the second enhancement layer into the more important window. Another option would be to fix $\Pi_1 = 0.065$, such that the reception constraint $\epsilon_{R,1}$ could be decreased to its minimum $\epsilon_{R,1}^{min} = 0.16$ (or approximately 450 kbps), that still satisfies the same performance constraint $\boldsymbol{P_{th}}$.

For the fixed value of $\Pi_1 = 0.065$ we optimize Γ_1 value inside the interval $[0.25, 0.68]$ with respect to the expected $PSNR$ of the j-th receiver class, $PSNR_{avg}^{(j)}$:

$$PSNR_{avg}^{(j)} = \sum_{i=0}^{L} P^{(j)}(i) \cdot PSNR(i), \qquad (2)$$

where L is the total number of layers (including the base layer denoted as layer 1), $PSNR(i)$ is the peak signal-to-noise ratio upon the complete recovery of i layers, averaged over all frames (Table 1), $PSNR(0) = 0$, and $P^{(j)}(i)$ is the probability that the first i layers are completely recovered by the j-th receiver class. Using the probability of complete recovery $P_1^{(j)}$ of the first window at the j-th receiver class, and the probability of recovery $P_{2,s}^{(j)}$ of the first s symbols of the second window at the j-th receiver class, we obtain:

$$P^{(j)}(i) = \begin{cases} 1 - P_1^{(j)} & i = 0 \\ P_1^{(j)} P_{2,k_i-k_1}^{(j)} (1 - P_{2,s_{i+1}}^{(j)}) & i = 1, \ldots, L-1 \\ P_1^{(j)} (1 - P_{2,K-k_1}^{(j)}) & i = L, \end{cases} \qquad (3)$$

where

$$P_{2,s}^{(j)} = (1 - y_{l,2}(\epsilon_{R,j}))^s. \qquad (4)$$

The average $PSNR$ values at the two receiver classes, $PSNR_{avg}^{(1)}$ and $PSNR_{avg}^{(2)}$, and the average $PSNR_{avg}$ across both receiver classes are presented in Figure 4. If we select Γ_1 as the value that maximizes $PSNR_{avg}$, a wide range of Γ_1 values between approximately $[0.3, 0.55]$ are equally good. Receiver classes $PSNRs$ demonstrate that the obtained solution set for EWF code parameters are "tuned" to match the EWF code performance to both receiver classes.

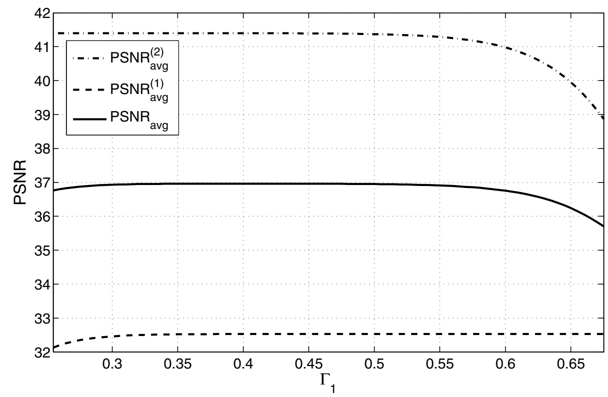

Fig. 4. Expected PSNRs in the solution interval of Γ_1 values.

6. CONCLUSIONS

In this paper, we analyzed the performance of the Expanding Window Fountain codes for scalable video streaming applications. Although the analysis can be applied on any number of receiver and data importance classes, for simplicity, we presented in detail the design of two window EWF codes. Through the code design example and its performance analysis, we illustrate very promising flexibility and efficiency of EWF codes in adapting the code at the video server side to receivers with heterogeneous reception capabilities.

7. REFERENCES

[1] M. Luby, "Lt codes," in *Proc. of the 43rd FOCS Symp.*, Vancouver, Canada, 2002, pp. 271–282.

[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Info. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[3] Q. Xu, V. Stanković, and Z. Xiong, "Wyner-ziv video compression and fountain codes for receiver-driven layered multicast," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 17, no. 7, pp. 901–906, 2007.

[4] J. Wagner, J. Chakareski, and P. Frossard, "Streaming of scalable video from multiple server using rateless codes," in *Proc. of the IEEE ICME*, Toronto, Canada, 2006.

[5] D. Sejdinović, D. Vukobratović, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection," in *41st Asilomar Conf.*, USA, 2007.

[6] N. Rahnavard, B. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property," *IEEE Trans. Info. Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[7] A. G. Dimakis, J. Wang, and K. Ramchandran, "Unequal growth codes: Intermediate performance and unequal error protection for video streaming," in *MMSP Workshop*, Crete, Greece, 2007.

[8] D. Vukobratović, V. Stanković, D. Sejdinović, L. Stanković (Fagoonee), and Z. Xiong, "Scalable data multicast using expanding window fountain codes," in *45th Allerton Conf.*, USA, 2007.