Expanding Window Fountain Codes for Unequal Error Protection

Dino Sejdinović, Dejan Vukobratović, Angela Doufexi, Vojin Šenk, and Robert J. Piechocki

***Abstract*—A novel approach to provide unequal error protection (UEP) using rateless codes over erasure channels, named Expanding Window Fountain (EWF) codes, is developed and discussed. EWF codes use a windowing technique rather than a weighted (non-uniform) selection of input symbols to achieve UEP property. The windowing approach introduces additional parameters in the UEP rateless code design, making it more general and flexible than the weighted approach. Furthermore, the windowing approach provides better performance of UEP scheme, which is confirmed both theoretically and experimentally.**

***Index Terms*—Asymptotic analysis, iterative decoding, maximum-likelihood decoding, rateless codes, unequal error protection.**

I. Introduction

FOUNTAIN codes, also called rateless codes, were investigated in [1] as an alternative to the automatic repeat-request (ARQ) schemes for reliable communication over lossy networks. They enable the transmitter to generate a potentially infinite stream of encoding symbols as random and equally important descriptions of the message block of finite length. A binary fountain code ensemble on the message block $\boldsymbol{x} = (x_1, x_2, \ldots, x_k) \in \mathbf{F}_2^k$ of k input symbols is defined by a linear map

$$\mathcal{F} : \boldsymbol{x} \mapsto (y_j)_{j \in \mathbf{N}}, \tag{1}$$

with

$$y_j = Y_j(\boldsymbol{x}) = \bigoplus_{i \in S_j} x_i, \tag{2}$$

$$S_j \subseteq N_k = \{1, 2, \ldots, k\},\ j \in \mathbf{N}.$$

Each output symbol y_j from the sequence $(y_j)_{j \in \mathbf{N}}$ is determined by the random and independent realization of the subset S_j as a modulo 2 sum, denoted by $\bigoplus$, of all the input symbols whose indices are in S_j. A fountain code ensemble is fully described by the probability mass function Υ on $\mathbf{F}_2^k$ which determines the sequence of functions $\{Y_j\}_{j \in \mathbf{N}}$ through realizations of the subsets $\{S_j\}_{j \in \mathbf{N}}$.

The first practical capacity approaching fountain codes, Luby Transform (LT) codes, were introduced in [2]. LT codes restrict Υ to depend only on the output symbol degrees, i.e., they assign the same probability to all the vectors in $\mathbf{F}_2^k$ of the same weight. Thus, they are described by a single distribution $\Omega_0, \Omega_1, \ldots \Omega_k$ on the set of possible weights $\{0, 1, \ldots, k\}$, called the output symbol degree distribution and denoted by its generator polynomial $\Omega(x) = \sum_{i=0}^{k} \Omega_i x^i$. For the appropriately selected sequence of output symbol degree distributions, such as robust soliton distributions [2], LT codes approach capacity on the binary erasure channel (BEC) of any erasure rate with encoding/decoding complexity of the order $O(k \log k)$, when decoded with the iterative belief propagation (BP) algorithm. Raptor codes [3] are a modification of LT codes obtained by precoding the input message block by a high rate low-density parity-check (LDPC) code, and by using a constant average output symbol degree distribution. Raptor codes were shown to have excellent performance in practice and linear encoding/decoding times. They are being adopted for large scale multimedia content delivery in practical systems, such as Multimedia Broadcast Multicast Services (MBMS) within 3GPP [4] and IP-Datacast (IPDC) within DVB-H [5].

LT and Raptor codes, as originally studied, provide equal error protection (EEP) for all of the input symbols, since after the output symbol degree has been selected, input symbols are drawn uniformly. However, there are cases where not all of the input symbols require the same protection. For example, in applications such as the transmission of video or images compressed with any of the numerous layered coders (MPEG, H.264...), certain data parts are considered to be more important. Additionally, in video-on-demand systems, a portion of data needs to be reconstructed prior to other parts. These applications, respectively, call for the coding schemes with unequal error protection (UEP) and unequal recovery time (URT) properties.

In this paper, we propose and investigate a novel class of fountain codes which can be used to provide UEP and URT properties by applying the idea of "windowing" the data set. We consider a predefined sequence of strictly increasing subsets of the data set, named windows, and assign a window to each output symbol with respect to a certain probability distribution. While forming an output symbol, the input symbols from the selected window are drawn uniformly, which makes this scheme easy to implement. The input symbols in the smallest window will have the strongest error protection, since they are contained in all possible windows and are considered in the forming of each output symbol. We show using both, analytical techniques and extensive simulations, that the windowing approach introduces additional freedom in the design of UEP rateless codes, thereby offering larger flexibility and better performance than the previously studied UEP fountain codes.

Paper approved by F. Fekri, the Editor for LDPC Codes and Applications of the IEEE Communications Society. Manuscript received November 26, 2007; revised August 9, 2008.

The material in this paper was presented in part at the 41st Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, Nov. 2007.

D. Sejdinović, A. Doufexi, and R. J. Piechocki are with the Centre for Communications Research, Department of Electrical and Electronic Engineering, University of Bristol, Bristol, UK (e-mail: {d.sejdinovic, a.doufexi, r.j.piechocki}@bristol.ac.uk).

D. Vukobratović and V. Šenk are with the Department of Communications and Signal Processing, University of Novi Sad, Novi Sad, Serbia (e-mail: {dejanv, ram_senk}@uns.ns.ac.yu).

Digital Object Identifier 10.1109/TCOMM.2009.09.070616

II. Related Work

In this section, a short overview of the recent work related to the windowing UEP fountain scheme is provided. We focus on recent contributions in fountain coding that provide UEP property or are based on windowing techniques.

Rahnavard et al. [6] studied a class of fountain codes which provide UEP and URT properties. In their work, the message block to be transmitted is partitioned into classes of different importance and input symbols from different classes are assigned different probabilities of being chosen after a degree of an output symbol has been selected, which produces a bias towards certain classes of symbols. The assignment of the probabilities is done in such a fashion that input symbols from the more important classes are more likely to be chosen in forming the output symbols, resulting in the UEP property. Therefore, this approach is a generalization of LT codes in which the neighbors of an output symbol are selected non-uniformly at random. We refer to this approach as the weighted approach.

Recently, different low-complexity approaches to fountain coding were studied, where the set of input symbols is divided into a number of overlapping subsets - windows, and only input symbols from a predetermined window can be used in forming each output symbol. To the best of our knowledge, Studholme and Blake were the first to utilize windowing approach in rateless codes, by introducing windowed erasure codes [7], where only a certain predetermined portion of the data set can be used in forming each output symbol. Their approach aims for EEP fountain codes with low encoding complexity and capacity achieving behavior assuming maximum-likelihood (ML) decoding, and is particularly suitable for short length codes. Targeting the real-time services such as multimedia streaming, the sliding window fountain codes were recently proposed in [8]. The sliding window fountain codes move the fixed-sized window forward during the encoding process, following the chronological ordering of data. For each window position, the number of generated encoding symbols is considerably smaller than the number of encoding symbols neccessary to successfully decode the window based on those symbols only, but since the consecutive windows overlap, successful decoding at the receiver is still possible.

In following section, we describe our windowing fountain approach for UEP applications.

III. Expanding Window Fountain Codes

A. EWF Codes: Generalization of LT Codes

We consider the transmission of data partitioned into blocks of k symbols over an erasure channel. For the sake of simplicity, symbol alphabet is set to $\mathbf{F}_2$. Let us assume that the numbers $s_1, s_2, \dots, s_r$, such that $s_1 + s_2 + \dots + s_r = k$, determine the partition of each block into classes of input symbols of different importance to the receiver, such that the first s_1 input symbols in a block form the first class, the next s_2 input symbols form the second class, etc. We further assume that the importance of classes decreases with the class index, i.e. that the i-th class is more important than the j-th class if $i < j$. This partition determines a sequence of strictly increasing subsets of the data set, which we call windows. The i-th window consists of the first $k_i = \sum_{j=1}^{i} s_j$ input symbols, and thus the most important symbols form the first window, whereas the entire block is the final r-th window. Note that the input symbols from the i-th class of importance belong to the i-th and all the subsequent windows, as illustrated in Fig. 1. We compactly describe the division into importance classes using the generating polynomial $\Pi(x) = \sum_{i=1}^{r} \Pi_i x^i$, where $\Pi_i = \frac{s_i}{k}$. In addition, it is useful to introduce $\Theta_i = \frac{k_i}{k} = \sum_{j=1}^{i} \Pi_j$ to our notation.

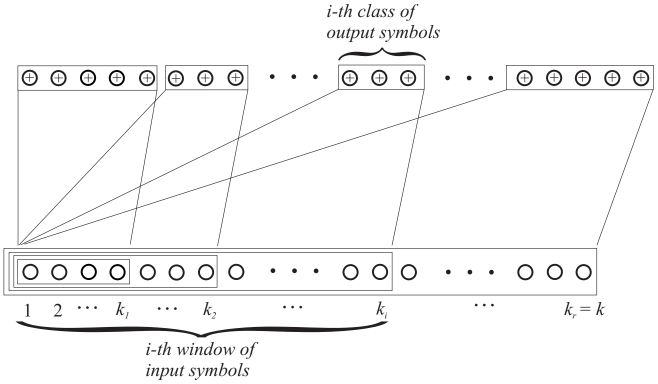

Fig. 1. Expanding window fountain codes.

In contrast to standard LT codes, we propose a scheme that assigns each output symbol to a randomly chosen window with respect to the window selection distribution $\Gamma(x) = \sum_{i=1}^{r} \Gamma_i x^i$, where Γ_i is the probability that the i-th window is chosen. Then, the output symbol is determined as if encoding is performed only on the selected window with an LT code of suitably chosen degree distribution. To summarize, EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \dots, \Omega^{(r)})$ is a fountain code which assigns each output symbol to the j-th window with probability Γ_j and encodes the chosen window using the LT code with distribution $\Omega^{(j)}(x) = \sum_{i=1}^{k_j} \Omega_i^{(j)} x^i$. In the case when $r = 1$, we obtain a standard LT code for equal error protection. Another extreme case occurs when $r = k$, where each input symbol is of different importance.

B. EWF Codes Probability Distribution on $\mathbf{F}_2^k$

Although the EWF codes utilize a slightly more complicated design with a number of additional parameters compared to LT codes, they belong to the class of fountain codes described by equations (1) and (2). Namely, the design parameters of an EWF code induce the probability distribution Υ on $\mathbf{F}_2^k$ which determines the sequence of functions $\{Y_j\}_{j \in \mathbf{N}}$. This is true for LT and Raptor codes as well, where for LT codes with the output degree distribution $\Omega(x)$, the induced distribution on $\mathbf{F}_2^k$ is given by

$$\Upsilon_{LT}(\boldsymbol{z}) = \frac{\Omega_{w(\boldsymbol{z})}}{\binom{k}{w(\boldsymbol{z})}}, \ \boldsymbol{z} \in \mathbf{F}_2^k, \tag{3}$$

where $w(\boldsymbol{z})$ denotes the Hamming weight of vector $\boldsymbol{z}$. The probability distribution Υ corresponding to an EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \dots, \Omega^{(r)})$ is given by

$$\Upsilon_{EW}(\boldsymbol{z}) = \sum_{l=c(\boldsymbol{z})}^{r} \Gamma_l \frac{\Omega_{w(\boldsymbol{z})}^{(l)}}{\binom{k_l}{w(\boldsymbol{z})}}, \ \boldsymbol{z} \in \mathbf{F}_2^k, \tag{4}$$

where $c(\boldsymbol{z}) = \min\{j \in N_r : z_i = 0, \ \forall i > k_j\}$ determines the smallest window in which non-zero elements of $\boldsymbol{z}$ are fully contained.

C. Asymptotic Degree Distributions of EWF Codes

As the starting point for the density evolution analysis, we derive the asymptotic degree distributions of EWF codes (as k tends to infinity). We assume EWF codes with a fixed reception overhead ε, i.e., with a total of $(1+\varepsilon)k$ output symbols collected at the receiver. The asymptotic degree distributions are derived for each of r different classes of input and output symbols.

The set of output symbol degree distributions is given by the code definition. We classify the set of output symbols in r classes of symbols associated to different windows. The asymptotic degree distribution of the output symbols in the j-th class is $\Omega^{(j)}(x)$. The average size of the j-th class is $\Gamma_j(1+\varepsilon)k$ output symbols and the average degree of output symbols in this class is equal to $\mu_j = \sum_i i\Omega_i^{(j)}$.

The set of input symbol degree distributions is given by the following Lemma:

Lemma 3.1: Asymptotic degree distribution of the input symbol nodes in the j-th class of importance is Poisson distribution

$$\Lambda^{(j)}(x) = \mathcal{P}\left((1+\varepsilon)\sum_{i=j}^{r}\frac{\mu_i\Gamma_i}{\Theta_i}\right). \tag{5}$$

One can obtain the set of distributions $\Lambda^{(j)}(x)$ starting from the distribution $\Lambda^{(r)}(x) = \sum_i \Lambda_i^{(r)} x^i = \sum_i \lambda_i^{(r)} x^i$, where $\lambda^{(j)}(x) = \sum_i \lambda_i^{(j)} x^i$ is the degree distribution of the input symbol nodes in the j-th window of size k_j, induced only by the edges connected to the output symbols in the j-th class. $\Lambda^{(r)}(x)$ asymptotically tends to the Poisson distribution[1] with the mean $\mu_r\Gamma_r(1+\varepsilon)$, denoted as $\mathcal{P}(\mu_r\Gamma_r(1+\varepsilon))$. Distributions $\Lambda^{(j)}(x)$, for $j < r$ are obtained by sequentially removing output symbol classes $r, r-1, \ldots, 2$ and their associated edges.

D. And-Or Tree Analysis of EWF Codes

The degree distributions derived in the previous section allow us to apply asymptotic and-or tree (density evolution) analysis on EWF codes. As a result, we obtain the expressions for asymptotic erasure probabilities after l iterations of the iterative BP decoding algorithm, for the input symbols in each of the input symbol classes. The original and-or tree analysis [9] is generalized in [6] for the weighted approach, where different classes of OR nodes in and-or trees are introduced, and the associated and-or tree lemma is derived. In a similar fashion, we further generalize the and-or tree construction by introducing different classes of AND nodes, and derive the corresponding version of an and-or tree lemma suitable for analysis of EWF codes.

In our setting, the generalized and-or tree $GT_{l,j}$ is constructed using r different classes of both AND and OR nodes. Let the root node of $GT_{l,j}$ belongs to the j-th class of OR nodes and the tree is expanded for $2l$ levels. Each AND and OR node from the m-th class has i children with probabilities $\beta_{i,m}$ and $\delta_{i,m}$, respectively. However, to analyze the EWF codes, we introduce a limitation that an AND node from the m-th class can only have OR node children belonging to the classes $\{1, 2, \ldots, m\}$, with the associated probabilities of choosing a child from the different OR classes being $\{q_1^{(m)}, q_2^{(m)}, \ldots, q_m^{(m)}\}$. Similarly, an OR node from the m-th class can only have AND node children from the classes $\{m, m+1, \ldots, r\}$, with the associated class probabilities $\{p_m^{(m)}, p_{m+1}^{(m)}, \ldots, p_r^{(m)}\}$. Let the nodes from the m-th class at the tree depth $2l$ be initialized as 0 with probability $y_{0,m}$, and 1 otherwise. It is assumed that OR nodes with no children have a value equal to 0, whereas AND nodes with no children have a value equal to 1. We state the following generalized version of the and-or tree lemma:

Lemma 3.2: Let $y_{l,j}$ be the probability that the root of an and-or tree $GT_{l,j}$ evaluates to 0. Then

$$y_{l,j} = \delta_j\left(1 - \sum_{i=j}^{r} p_i^{(j)} \beta_i\left(1 - \sum_{m=1}^{i} q_m^{(i)} y_{l-1,m}\right)\right) \tag{6}$$

where $\delta_j(x) = \sum_i \delta_{i,j} x^i$ and $\beta_j(x) = \sum_i \beta_{i,j} x^i$.

We skip the proof of our version of generalized and-or tree lemma since it closely follows the proof of the original and-or tree lemma [9].

From the asymptotic degree distributions of EWF codes and the design rules for their construction, we can derive polynomials $\delta_m(x)$ and $\beta_m(x)$ and the probabilities $\{q_1^{(m)}, q_2^{(m)}, \ldots, q_m^{(m)}\}$ and $\{p_m^{(m)}, p_{m+1}^{(m)}, \ldots, p_r^{(m)}\}$, for each class m of input and output symbols. Similar to the derivation in [6], $\beta_{i,j}$, which is the probability that the output symbol connected with a randomly selected edge has degree $i+1$ given that it belongs to the class j, equals $\beta_{i,j} = \frac{(i+1)\Omega_{i+1}^{(j)}}{\Omega^{(j)\prime}(1)}$, i.e., that $\beta_j(x) = \frac{\Omega^{(j)\prime}(x)}{\Omega^{(j)\prime}(1)}$. Similarly, it can be shown that the probability $\delta_{i,j}$ that the variable node connected with a randomly selected edge has degree $i+1$, given that it belongs to the class j, equals $\delta_{i,j} = \frac{(i+1)\Lambda_{i+1}^{(j)}}{(1+\varepsilon)\sum_{l=j}^{r}\frac{\mu_l\Gamma_l}{\Theta_l}}$, i.e., that $\delta_j(x) = e^{(1+\varepsilon)\sum_{l=m}^{r}\frac{\mu_l\Gamma_l}{\Theta_l}(x-1)}$. It is easy to show that for the class m input symbols, the probability of having class j output symbol as a children, $m \le j \le r$, equals $p_j^{(m)} = \frac{\frac{\mu_j\Gamma_j}{\Theta_j}}{\sum_{i=m}^{r}\frac{\mu_i\Gamma_i}{\Theta_i}}$. Similarly, the class m output symbols have the class j input symbol child, $1 \le j \le m$, with probability $q_j^{(m)} = \frac{s_j}{k_m}$.

Substituting these results into *Lemma* 3.2, we obtain the erasure probability evolution for input nodes of EWF codes decoded iteratively, as stated in the following lemma.

Lemma 3.3: For an EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$, the probability $y_{l,j}$ that the input node of class j is not recovered after l iterations of BP algorithm for the reception overhead ε is

$$y_{0,j} = 1, \tag{7}$$
$$y_{l,j} = e^{\left(-(1+\varepsilon)\sum_{i=j}^{r}\frac{\Gamma_i}{\sum_{t=1}^{i}\Pi_t}\Omega^{(i)\prime}\left(1-\frac{\sum_{m=1}^{i}\Pi_m y_{l-1,m}}{\sum_{t=1}^{i}\Pi_t}\right)\right)}.$$

[1] The convergence towards the Poisson distribution is under the same conditions as given in [6], Section III.

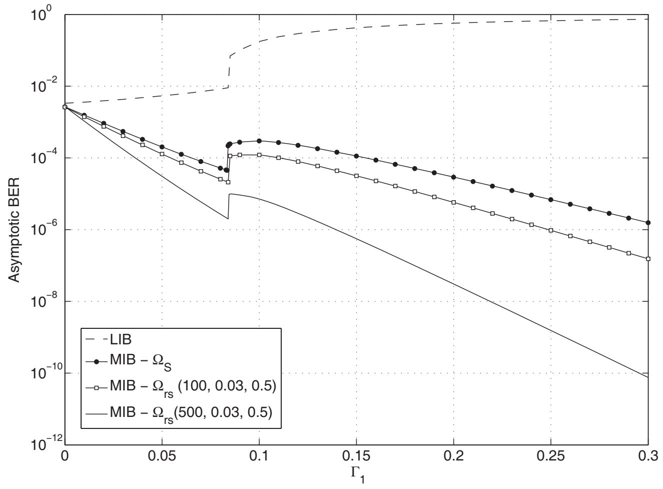

Fig. 2. Asymptotic analysis of BER versus Γ_1 for EWF codes.

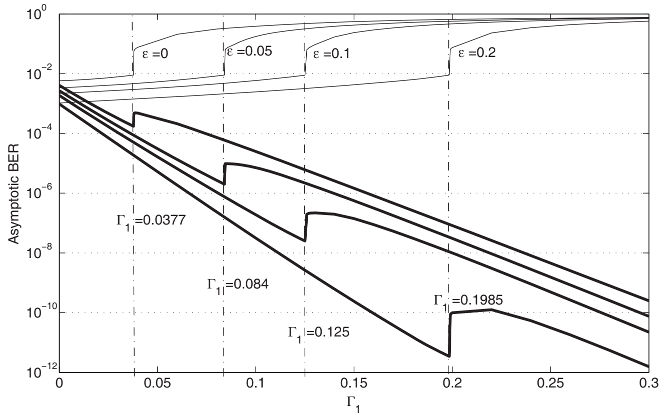

Fig. 3. Optimizing Γ_1 parameter for various overheads.

The particularly simple and important scenario is when the set of input symbols is divided in two importance classes, the class of more important bits (MIB) and less important bits (LIB). In the following, we provide a special case of *Lemma* 3.3 for this scenario. Additionally, we compare the obtained results with the UEP fountain codes [6] and point out to the benefits of using the windowing instead of the weighted approach.

E. EWF Codes With Two Importance Classes

For the special case of two importance classes of input symbols, we use *Lemma* 3.3 to track the asymptotic erasure probabilities of MIB and LIB. For an EWF code $\mathcal{F}_{EW}(\Pi_1 x+\Pi_2 x^2, \Gamma_1 x+\Gamma_2 x^2, \Omega^{(1)}, \Omega^{(2)})$ we obtain the expressions for the erasure probabilities of MIB and LIB after l iterations, $y_{l,MIB}$ and $y_{l,LIB}$, respectively as in (8) and (9), for $l \geq 1$ and $y_{0,MIB} = y_{0,LIB} = 1$.

We select the parameters of the erasure probabilities formulae (8) and (9) in order to compare our results with the results obtained in [6]. Therefore, we analyze $\mathcal{F}_{EW}(0.1x + 0.9x^2, \Gamma_1 x + (1-\Gamma_1)x^2, \Omega^{(1)}, \Omega^{(2)})$ EWF codes with the reception overhead $\varepsilon = 0.05$ and the same degree distribution applied on both windows, adopted from [6] (originally from [3]):

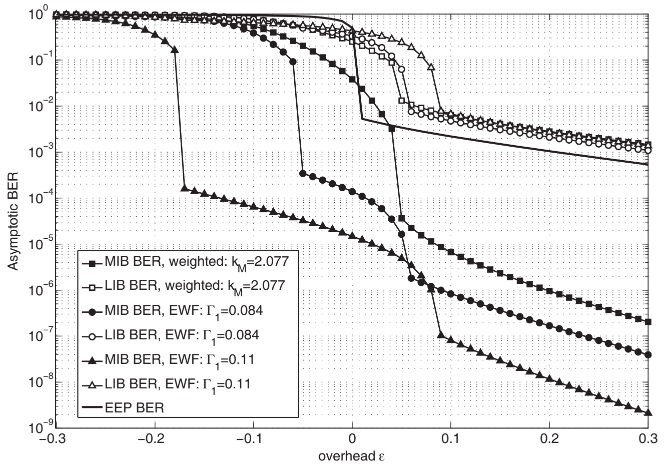

Fig. 4. Asymptotic analysis of BER versus the overhead ε.

$$\begin{aligned}\Omega^{(1)}(x) = \Omega^{(2)}(x) = 0.0080x + 0.4936x^2 +\\ 0.1662x^3 + 0.0726x^4 + 0.0826x^5 + 0.0561x^8 +\\ 0.0372x^9 + 0.0556x^{19} + 0.0250x^{64} + 0.0031x^{66}. \qquad (10)\end{aligned}$$

Fig. 2 shows the dependence of the asymptotic erasure probabilities, $y_{\infty,MIB}$ and $y_{\infty,LIB}$, on the first window selection probability Γ_1. Note that by varying Γ_1 we change the probability of the input symbol selection from different input symbol classes, similarly as it is explicitly done with the parameter k_M used in [6]. For an extreme case of $\Gamma_1 = 0$, we have the EEP fountain codes, whereas by increasing Γ_1 we progressively add protection to the MIB class.

It is interesting to note that the desirable point of local minimum of $y_{\infty,MIB}$ (where $y_{\infty,LIB}$ is still not significantly deteriorated) occurs in our case for the first window selection probability $\Gamma_1 = 0.084$, and is equal to $y_{\infty,MIB}^{(min)} = 4.6 \cdot 10^{-5}$. The equivalent point in [6] occurs for $k_M = 2.077$ where $y_{\infty,MIB}^{(min)} = 3.8 \cdot 10^{-5}$, which is a slightly better performance than in the EWF case. This small degradation suggests the negative effect of the windowing approach, due to the fact that the output symbols based on the MIB window do not contain any information about LIB. However, in this example we did not exploit the positive side of the EWF codes, namely, to use a different (stronger) degree distribution on the smaller (MIB) window. Although the stronger distributions can be the subject of further optimization, in this work we use a simple ad hoc method of "enhancing" the strength of the MIB window distribution, by applying the "truncated" robust soliton distribution, i.e., the robust soliton distribution up to a certain degree k_{rs} generally much smaller than the length of the sequence being encoded. In other words, we use the robust soliton distribution $\Omega_{rs}(k_{rs}, \delta, c)$ [2] with a constant value of maximum degree k_{rs} and apply it on the MIB window (note that the size of the MIB window $\Pi_1 k$ asymptotically tends to infinity and thus computational complexity remains linear in k). The results for $k_{rs} = 100$ and $k_{rs} = 500$ are presented in Fig. 2. The performance improvement of the EWF approach is obvious, reaching an order of magnitude lower local minimum of $y_{\infty,MIB}^{(min)} = 2.2 \cdot 10^{-6}$. We note that an important advantage

$$y_{l,MIB} = \exp\left(-(1+\varepsilon)\left(\frac{\Gamma_1}{\Pi_1}\Omega^{(1)'}(1-y_{l-1,MIB}) + \Gamma_2\Omega^{(2)'}(1-\Pi_1 y_{l-1,MIB} - \Pi_2 y_{l-1,LIB})\right)\right) \tag{8}$$

$$y_{l,LIB} = \exp\left(-(1+\varepsilon)\Gamma_2\Omega^{(2)'}(1-\Pi_1 y_{l-1,MIB} - \Pi_2 y_{l-1,LIB})\right) \tag{9}$$

of the proposed technique lies in its possibility to employ different degree distributions in different windows and thus produce performance improvements for a more important class of data without sacrificing performance for a less important class of data.

Fig. 3 shows the variations of the local minima of $y_{\infty,MIB}$ and $y_{\infty,LIB}$, when $\Omega_{rs}(k_{rs} = 500, \delta = 0.5, c = 0.03)$ distribution is applied on the MIB window, for the values of the reception overhead equal to $\varepsilon = 0, 0.05, 0.1$, and 0.2. The selection probability of the MIB window, Γ_1, that corresponds to the points of local minima of $y_{\infty,MIB}$ and $y_{\infty,LIB}$, is presented for each of the ε values. We note that as the reception overhead ε grows, there is an opportunity for significant decrease of $y_{\infty,MIB}$ by increasing the first window selection probability Γ_1, without notable loss of performance on LIB compared with the performance of EEP fountain codes. Fig. 4 illustrates the asymptotic erasure probability curves of MIB and LIB classes as a function of the reception overhead ε. We compare the EWF code with the $\Omega_{rs}(k_{rs} = 500, \delta = 0.5, c = 0.03)$ distribution applied on the MIB window, with the weighted UEP fountain codes from [6]. For the EWF code, we use the first window selection probability $\Gamma_1 = 0.084$ which is optimized for the reception overhead $\varepsilon = 0.05$ (Fig. 3), whereas for the weighted UEP fountain codes we use parameter value $k_M = 2.077$ optimized for the same reception overhead. Fig. 4 clearly shows that the EWF codes show stronger URT and UEP properties than the corresponding weighted codes. It is significant to note that in most cases MIB symbols can be decoded well before the reception of k output symbols, due to the fact that the decoder makes use of the packets which contain only MIB-information. This manifests in two "decoding avalanches" in the erasure probability curves of the EWF codes. The URT properties become more notable as we increase Γ_1 with a small loss in LIB performance. This is illustrated in Fig. 4 with the example of the EWF code with the same design parameters, except that its first window selection probability is increased to the value $\Gamma_1 = 0.11$.

IV. Lower and Upper Bounds on the ML Decoding of EWF Codes

A simple lower bound on the bit error rate of EWF codes under the ML decoding can be calculated as a probability that an input symbol node in the fixed class is not adjacent to any of the output symbol nodes. The probability that the input symbol node in the i-th class is not adjacent to some output symbol node in the j-th class is $1-\frac{\mu_j}{k_j}$, where μ_j is the average degree of the distribution $\Omega^{(j)}(x)$, provided that $j \geq i$. After averaging over the window selection distribution $\Gamma(x)$, we obtain the lower bound on the ML decoding of the input symbols in the i-th importance class of $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ as

$$p_i^{ML}(\varepsilon) \geq (1 - \sum_{j=i}^{r} \frac{\Gamma_j \mu_j}{k_j})^{k(1+\varepsilon)}. \tag{11}$$

The upper bound on the bit error rate of the input symbols from different importance classes of EWF codes is derived as the sum of probabilities that an arbitrary vector in $\mathbf{F}_2^k$, with a non-zero element corresponding to the input symbol node in the i-th class, belongs to the the dual space of the punctured generator matrix G of the EWF code. The upper bound is provided in the following lemma:

Lemma 4.1: The bit error rate of the input symbols in the i-th importance class of an EWF code $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$, for the reception overhead ε, under the ML decoding, is upper bounded by

$$p_i^{ML}(\varepsilon) \leq \min\Bigg\{1, \sum_{t_r=1}^{k} \cdots \sum_{t_i=1}^{t_{i+1}} \sum_{t_{i-1}=0}^{t_i-1} \cdots \sum_{t_1=0}^{t_2} \prod_{p=1}^{r} \binom{k_p - k_{p-1} - \delta(p-i)}{t_p - t_{p-1} - \delta(p-i)} \cdot \left(\sum_{j=1}^{r} \Gamma_j \sum_{d=1}^{k_j} \Omega_d^{(j)} \frac{\sum_{s=0}^{\lfloor d/2 \rfloor} \binom{t_j}{2s}\binom{k_j-t_j}{d-2s}}{\binom{k_j}{d}}\right)^{(1+\varepsilon)k}\Bigg\}. \tag{12}$$

We skip the detailed proof of the lemma as it is a straightforward generalisation of the proof of Lemma 8 in [6]. It suffices to fix the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_k) \in \mathbf{F}_2^k$ such that $x_l = 1$, $k_{i-1} < l \leq k_i$, denote its weight in the j-th window by t_j, apply Lemma 8 in [6] for a fixed window j and average over the window selection distribution. What remains is to count the possible vectors $\boldsymbol{x} \in \mathbf{F}_2^k$ with the "window-weights" equal to $t_1, \ldots, t_r$ such that $x_l = 1$, and sum over all allowed choices for $t_1, \ldots, t_r$.

Fig. 5 represents the bounds on the ML decoding for $r = 2$, $k = 500$, $k_1 = 50$, $\Gamma_1 = 0.11$, and $\Omega^{(2)}$ as given in (10). The lower and upper bound become tight as the reception overhead increases. We obtain similar results as in [6] when $\Omega^{(1)}$ is the robust soliton distribution $\Omega_{rs}(k_{rs} = 50, \delta = 0.5, c = 0.03)$. As before, by modifying the output degree distribution on the smaller window we can strengthen the protection of MIB symbols and decrease the MIB bounds, while the LIB bounds remain effectively unchanged. For example, if $\Omega^{(1)}$ is set to the robust soliton distribution $\Omega_{rs}(k_{rs} = 50, \delta = 0.2, c = 0.03)$, the bounds on the ML decoding decrease as shown in Fig. 4. This illustrates how the UEP property of EWF codes may be improved by adapting distribution $\Omega^{(1)}$.

V. Simulation Results

In order to verify the results of the asymptotic and-or tree analysis developed in Section III D, we performed simulations

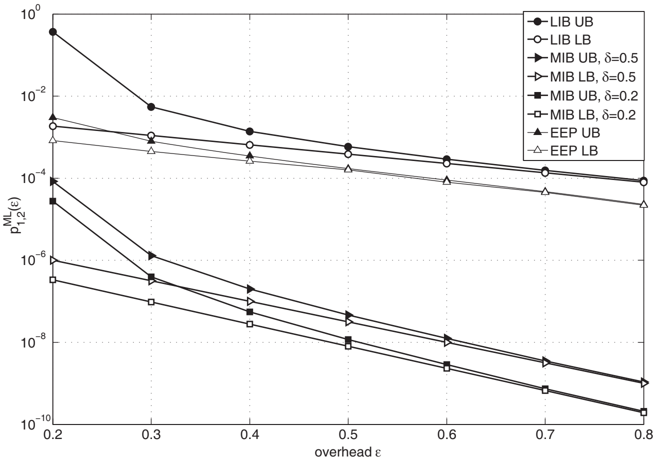

Fig. 5. Upper and lower bounds for the ML decoding of EWF codes.

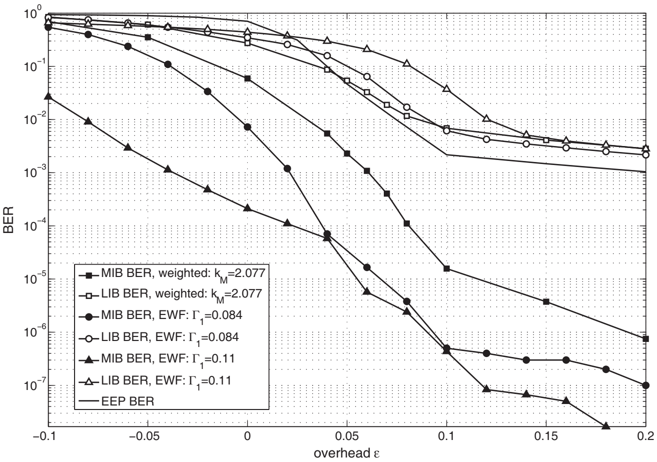

Fig. 6. Simulation results for $k = 5000$.

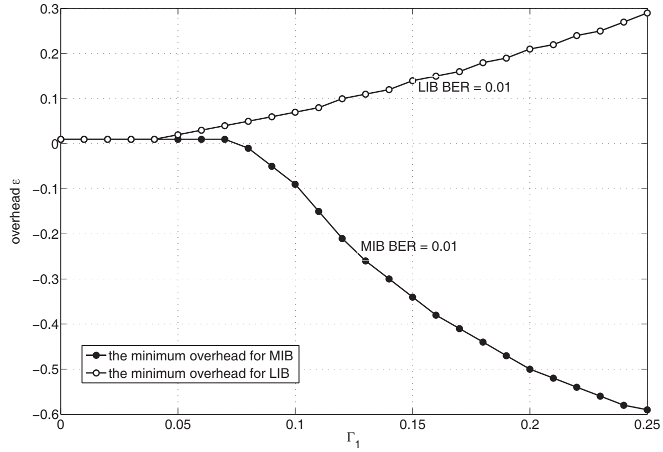

Fig. 7. The minimum overheads necessary to reach BER of 10^{-2} for both MIB and LIB symbols.

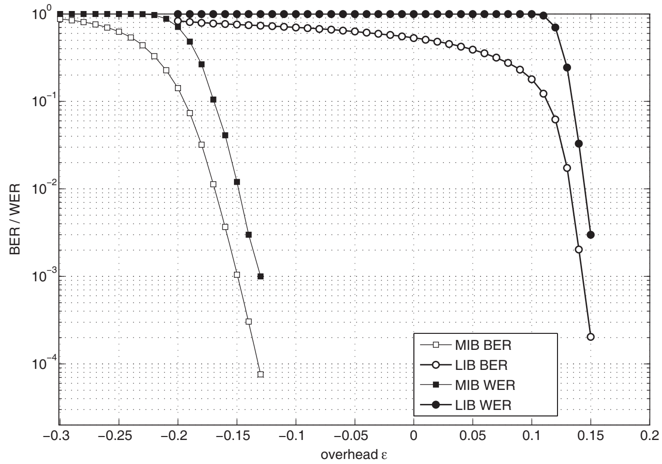

Fig. 8. The simulation results for precoded EWF codes.

to determine the BER performance of EWF codes with two importance classes. We assume that the MIB class contains 500 input symbols, out of the total number of $k = 5000$ input symbols. The simulations are performed for the same codes for which the asymptotic results on the BER performances are analyzed and presented in Fig. 4: the weighted fountain codes with parameter $k_M = 2.077$ and EWF codes with parameter $\Gamma_1 = 0.084$ and $\Gamma_1 = 0.11$. For EWF codes, the stronger degree distribution $\Omega_{rs}(k_{rs} = 500, \delta = 0.5, c = 0.03)$ is applied on the MIB window. At the receiver side, iterative LT decoding is assumed. Fig. 6 demonstrates that the simulated BER performance closely corresponds to the results predicted by the asymptotic analysis. Also, the results clearly show that EWF codes with the parameter $\Gamma_1 = 0.084$ outperform the weighted codes with parameter $k_M = 2.077$ [6] in terms of MIB BER. Increase in Γ_1, i.e., more frequent selection of the MIB window, further decreases MIB BER but introduces slight deterioration in terms of LIB BER.

To perform with linear encoding/decoding complexity, the EWF code ensembles utilize the output degree distributions of constant average degree. However, it is well known that the constant average degree distributions of LT codes result in a high error floor of their BER performance curves, due to the input symbols which remain "uncovered" by the output symbols. This problem is solved with the introduction of Raptor codes where LT codes are precoded using good high-rate error correcting codes such as LDPC codes [3]. In the following, we consider precoding EWF codes using high-rate LDPC codes. We perform separate precoding of different classes of input symbols, i.e., during the precoding process the input symbols of the i-th importance class are encoded using the high-rate LDPC code corresponding to their importance class, and the obtained codeword represents a new set of input symbols of the i-th importance class. Using precoding that separately precodes different importance classes, the content of each importance class can be recovered at the receiver side using the iterative decoder that operates simultaneously on both, the EWF part of the code graph, and the LDPC code graphs associated with each of the importance classes. Additionally, this design is useful since it allows for independent calculations of the overhead values for different importance classes in such a way that a full recovery of symbols of different importance classes is asymptotically guaranteed.

In the following, we assume that precoding is performed over EWF codes with two importance classes. Furthermore, we assume that outer high-rate LDPC codes over both MIB and LIB classes are capable of correcting the average erasure

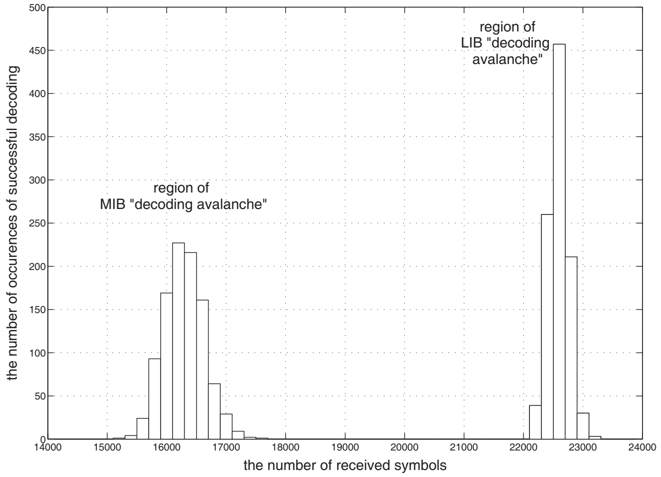

Fig. 9. The histograms of the number of received symbols necessary for the successful decoding of MIB and LIB symbols.

rate (BER) of up to 10^{-2}. Therefore, we expect the inner EWF code to provide the protection of precoded input symbols of both classes until their erasure performances reach the value of 10^{-2}. This performance level is attained at considerably lower overheads for the symbols of MIB class. Fig. 7 illustrates this fact, where the minimum overheads of EWF codes necessary to reach BER of 10^{-2} for both MIB and LIB are given as a function of the parameter Γ_1. Hence, it is possible to use the parameter Γ_1 in order to set the desired overheads for the two importance classes. In our simulation scenario, for the input block length of $k = 20000$, we select the value of $\Gamma_1 = 0.12$ in order to reach the BER level of 10^{-2} with the negative overhead $\varepsilon_M \sim -0.2$ for the MIB class, and with the small positive overhead of $\varepsilon_L \sim 0.1$ for the LIB class. Both MIB and LIB classes are precoded using concatenated Hamming/LDPC codes, adopted in [4] for precoding performed in Raptor codes. The achieved bit error rates and word error rates for MIB and LIB classes are presented in Fig. 8, where the effect of precoding is apparent in the region close to the desired overheads for both importance classes. Additionally, the histograms of the numbers of symbols necessary to decode the MIB class and the LIB class are presented in Fig. 9. These results confirm the strong URT property provided by precoded EWF codes.

VI. Conclusion

LT and Raptor codes are attractive forward error correction solution for multicasting data over erasure channels due to their rateless property. However, the equal error protection for all input symbols is also inherent in their construction, while many multicasting applications benefit from stronger error protection for a certain portion of input symbols. This is the case for multicasting image or video files compressed with any of the numerous layered or scalable coders, where input symbols are divided into a base layer, the set of the most important symbols, and a number of enhancement layers which progressively improve image or video quality. Hence, it is worthwhile to consider fountain coding techniques which offer unequal error protection (UEP) and unequal recovery time (URT) properties. In this paper, we presented an alternative way to construct such coding techniques which succeed in outperforming previously studied weighted UEP rateless code design.

EWF codes, as a rather straightforward generalization of LT codes, allow detailed asymptotic analysis of their performance and theoretical study does confirm their advantage compared to the weighted approach. A rather intuitive explanation exists for the advantage of windowing approach – the decoder can make use of the set of encoding symbols connected only to a smaller window in order to produce an early “decoding avalanche” which corrects most of the erasures in that window. We also show that by careful choice of the separate precoding scheme of different importance classes, this “decoding avalanche” can be utilized to perform full decoding of the window. This URT property can be viewed in a context of the delay-constrained multicast as well. Namely, in such applications, users with severe channel conditions cannot make use of the ratelessness of fountain codes and thus have access to a number of encoding symbols with very limited or even negative overhead. By application of EWF codes combined with the scalable video coders, it is possible to guarantee successful reception of the base layer even to users with severe channel conditions. Preliminary investigations of such scalable multicast transmission with EWF codes are presented in [10], [11] and analysis of further practical realizations of such schemes is a part of the ongoing investigation.

Acknowledgment

D. Sejdinovic and R.J. Piechocki would like to thank Toshiba Telecommunications Research Laboratory and its directors for supporting this work.

References

[1] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data,” in *Proc. ACM SIGCOMM*, Sept. 1998, pp. 56–67.

[2] M. Luby, “LT codes,” in *Proc. 43rd IEEE Symp. Foundations Computer Science (FOCS)*, Nov. 2002, pp. 271–282.

[3] A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[4] “Universal Mobile Telecommunications System (UMTS): Mobile Broadcast/Multicast Service (MBMS): Protocols and Codecs,” 3GPP TS 26.346, Version 6.3.0, Release 6, Dec. 2005.

[5] “Digital Video Broadcasting (DVB): Transmission System for Handheld Terminals (DVB-H),” ETSI EN 302 304 V1.1.1, 2004.

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

[7] C. Studholme and I. Blake, “Windowed erasure codes,” in *Proc. Intl. Symp. Inform. Theory (ISIT)*, July 2006, pp. 509–513.

[8] M. C. O. Bogino, P. Cataldi, M. Grangetto, E. Magli, and G. Olmo, “Sliding-window digital fountain codes for streaming multimedia contents,” in *Proc. Intl. Symp. Circuits Systems (ISCAS)*, May 2007, pp. 3467–3470.

[9] M. Luby, M. Mitzenmacher, and A. Shokrollahi, “Analysis of random processes via and-or tree evaluation,” in *Proc. 9th SIAM Symp. Discrete Algorithms (SODA)*, Jan. 1998, pp. 364–373.

[10] D. Vukobratović, V. Stanković, D. Sejdinović, L. (Fagoonee) Stanković, and Z. Xiong, “Scalable data multicast using expanding window fountain codes,” in *Proc. 45th Allerton Conf. Commun., Control, Computing*, Sept. 2007.

[11] D. Vukobratović, V. Stanković, D. Sejdinović, L. (Fagoonee) Stanković, and Z. Xiong, “Expanding window fountain codes for scalable video multicast,” in *Proc. IEEE Intl. Conf. Multimedia Expo. (ICME)*, June 2008.