Peer reviewed version

Link to published version (if available):
10.1109/ACSSC.2007.4487375

Link to publication record in Explore Bristol Research
PDF-document

Expanding Window Fountain Codes for Unequal Error Protection

Dino Sejdinović [†], Dejan Vukobratović [‡], Angela Doufexi [†], Vojin Šenk [‡], Robert J. Piechocki [†]

[†]*Centre for Communications Research, Department of Electrical and Electronic Engineering, University of Bristol, UK*
{d.sejdinovic, a.doufexi, r.j.piechocki}@bristol.ac.uk
[‡]*Department of Communications and Signal Processing, University of Novi Sad, Serbia*
{dejanv, ram_senk}@uns.ns.ac.yu

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.

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. In the case of a binary fountain code on the message block $\boldsymbol{x} = (x_1, x_2, \ldots, x_k) \in \mathbf{F}_2^k$ of k input symbols, each encoding symbol $y_j \in \mathbf{F}_2$, $j \in \mathbf{N}$ is generated independently as a scalar product $y_j = \mathbf{r}_j \cdot \mathbf{x}$, where $\mathbf{r}_j$ is the j-th realization of a random variable R on $\mathbf{F}_2^k$. Thus, one can describe a fountain code by the probability mass function of random variable R.

The first practical capacity achieving fountain codes, Luby Transform (LT) codes, were introduced in [2]. LT codes assign the same probability to all the vectors in $\mathbf{F}_2^k$ of the same weight, and are thus described by a single distribution Ω on the set of possible weights $\{0, 1, 2, \ldots, k\}$, i.e., by the output symbol degree distribution. For the appropriately selected output symbol degree distribution, the encoding and decoding complexity of LT codes is of the order $O(k \log k)$ if the suboptimal iterative belief propagation decoding algorithm is used. 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 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 input symbols. 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 image files 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 will start by pointing to the relevant related work on UEP fountain codes and windowing techniques used in the construction of fountain codes.

II. RELATED WORK

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 subsets of different importance and probabilities of selecting input symbols from different subsets are assigned. This is done in such a fashion that input symbols from the more important subsets 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 to 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]. Their approach aims for EEP fountain codes with low encoding complexity and capacity achieving behavior assuming maximum-likelihood 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], which move the fixed-sized window forward during the encoding process, following the chronological ordering of data.

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 $s_1, s_2, \ldots, s_r$, such that $s_1 + s_2 + \cdots + s_r = k$, 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 chronological ordering of the symbols, 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 and 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. 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.

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 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 $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ 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.

B. 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 $\mu_j = \sum_i i\Omega_i^{(j)}$.

To derive the set of input symbol degree distributions, we use the division of input symbols into r importance classes of size $\{s_1, s_2, \ldots, s_r\}$. Each input symbol class is described by the corresponding degree distribution from the set $\{\Lambda^{(1)}, \ldots, \Lambda^{(r)}\}$, where $\Lambda^{(m)}(x) = \sum_i \Lambda_i^{(m)} x^i$. The coefficients $\Lambda_i^{(m)}$ can be calculated from the set of input symbol degree distributions $\{\lambda^{(m)}(x), \lambda^{(m+1)}(x), \ldots, \lambda^{(r)}(x)\}$, 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. The coefficients $\lambda_i^{(j)}$ can be found as

$$\lambda_i^{(j)} = \binom{\mu_j \Gamma_j (1 + \varepsilon)k}{i} \left(\frac{1}{k_j}\right)^i \left(1 - \frac{1}{k_j}\right)^{\mu_j \Gamma_j (1+\varepsilon)k - i}, \tag{1}$$

resulting in the following expression for $\Lambda_i^{(m)}$

$$\Lambda_i^{(m)} = \sum_{S_i} \prod_{j=m}^{r} \lambda_{i_j}^{(j)}, \tag{2}$$

where $S_i = \{(i_m, i_{m+1}, \ldots, i_r) : \sum_{j=m}^{r} i_j = i\}$.

In general, using expressions (1) and (2) to obtain $\Lambda^{(j)}(x)$ leads to tedious calculations. However, it is easy to obtain the set of distributions $\Lambda^{(j)}(x)$ starting from the distribution $\Lambda^{(r)}(x) = \sum_i \Lambda_i^{(r)} x^i = \sum_i \lambda_i^{(r)} x^i$, since this distribution asymptotically tends to the Poisson distribution[1] with the mean $\mu_r \Gamma_r (1 + \varepsilon)$, denoted as $\mathcal{P}(\mu_r \Gamma_r (1 + \varepsilon))$. By sequentially removing input symbol classes and their associated edges, starting from the least important r-th class of input symbols, one can easily obtain the remaining set of the asymptotic input symbol degree distributions as the following set of Poisson distributions

$$\Lambda^{(j)}(x) = \mathcal{P}\left((1 + \varepsilon) \sum_{i=j}^{r} \frac{\mu_i \Gamma_i}{\Theta_i}\right) \tag{3}$$

The ensemble of EWF codes $\mathcal{F}_{EW}(\Pi, \Gamma, \Omega^{(1)}, \ldots, \Omega^{(r)})$ with a fixed reception overhead ε is asymptotically described with the number of windows r, polynomials $\Pi(x)$ and $\Gamma(x)$, and the set of degree distributions $(\Lambda^{(i)}(x), \Omega^{(i)}(x))$.

C. 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 message-passing 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, introducing different classes of AND nodes, and derive the corresponding version of an and-or tree lemma suitable for analysis of EWF codes.

---

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

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 AND children 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 3.1: Let $y_{l,j}$ be the probability that the root of an and-or tree $GT_{l,j}$ evaluates to 0. Then

$$y_{l,j} = \delta_j \left(1 - \sum_{i=j}^{r} p_i^{(j)} \beta_i \left(1 - \sum_{m=1}^{i} q_m^{(i)} y_{l-1,m} \right) \right) \quad (4)$$

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 $\beta_{i,j} = \frac{(i+1)\Omega_{i+1}^{(j)}}{\Omega'^{(j)}(1)}$, i.e., that $\beta_j(x) = \frac{\Omega'^{(j)}(x)}{\Omega'^{(j)}(1)}$. 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, $m \leq j \leq r$, 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, $1 \leq j \leq m$, with probability $q_j^{(m)} = \frac{s_j}{k_m}$.

Substituting these results into *Lemma* 3.1, we obtain the erasure probability evolution for input nodes of EWF codes decoded iteratively, as stated in the following lemma.

Lemma 3.2: 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 message-passing algorithm for the reception overhead ε is

$$y_{0,j} = 1 \quad (5)$$
$$y_{l,j} = e^{\left(-(1+\varepsilon)\sum_{i=j}^{r} \frac{\Gamma_i}{\sum_{t=1}^{i} \Pi_t} \Omega'^{(i)}\left(1 - \frac{\sum_{m=1}^{i} \Pi_m y_{l-1,m}}{\sum_{t=1}^{i} \Pi_t}\right)\right)}.$$

D. EWF Codes with Two Importance Classes

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). We use *Lemma* 3.2 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 (6) and (7), for $l \geq 1$ and $y_{0,MIB} = y_{0,LIB} = 1$.

We select the parameters of the erasure probabilities formulae (6) and (7) 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 output symbol degree distribution Ω_S applied on both windows, adopted from [6] (originally from [3]):

$$\begin{aligned}
\Omega^{(1)}(x) = \ & \Omega^{(2)}(x) = 0.007969x + 0.493570x^2 + \\
& + 0.166220x^3 + 0.072646x^4 + 0.082558x^5 + \\
& + 0.056058x^8 + 0.037229x^9 + 0.055590x^{19} + \\
& + 0.025023x^{64} + 0.003135x^{66} \quad (8)
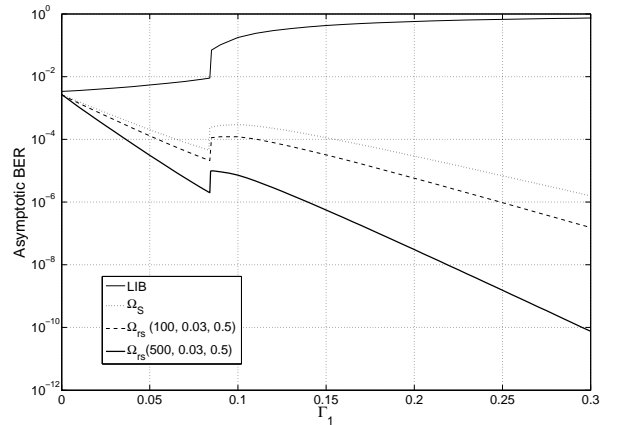\end{aligned}$$

Fig. 1. Asymptotic analysis of BER versus Γ_1 for EWF codes

Figure 1 shows the dependence of the asymptotic erasure probabilities, $y_{\infty,MIB}$ (dotted line) and $y_{\infty,LIB}$ (thin line), 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,

$$y_{l,MIB} = \exp\left(-(1+\varepsilon)\left(\frac{\Gamma_1}{\Pi_1}\Omega^{'(1)}(1-y_{l-1,MIB}) + \Gamma_2\Omega^{'(2)}(1-\Pi_1 y_{l-1,MIB} - \Pi_2 y_{l-1,LIB})\right)\right) \tag{6}$$

$$y_{l,LIB} = \exp\left(-(1+\varepsilon)\Gamma_2\Omega^{'(2)}(1-\Pi_1 y_{l-1,MIB} - \Pi_2 y_{l-1,LIB})\right) \tag{7}$$

whereas by increasing Γ_1 we progressively add protection to the MIB class.

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. In this work, we use a simple method of "enhancing" the strength of the MIB window distribution, by applying the "truncated" robust solition distribution $\Omega_{rs}(k_{rs},\delta,c)$ [2] with a constant value of k_{rs} (note that the size of the MIB window $\Pi_1 k$ asymptotically tends to infinity). The results for $k_{rs} = 100$ (dashed line) and $k_{rs} = 500$ (thick line) are presented in Figure 1. The performance improvement of the EWF approach is obvious, reaching an order of magnitude lower local minimum of $y_{\infty,MIB}^{(min)} = 2.2 \cdot 10^{-6}$.
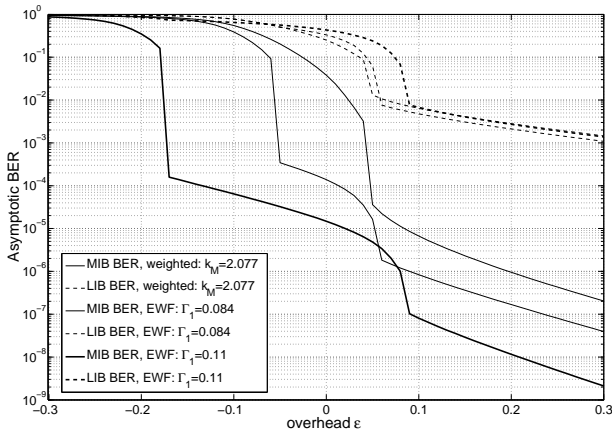

Fig. 2. Asymptotic analysis of BER versus the overhead ε

Figure 2 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

$\varepsilon = 0.05$, whereas for the weighted UEP fountain codes we use parameter value $k_M = 2.077$ optimized for the same reception overhead. Figure 2 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 Figure 2 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 maximum likelihood decoding can be calculated for each class of input symbol nodes separately, as a probability that an input symbol node is not adjacent to any of the output symbol nodes. Let us consider the input symbol nodes in the i-th class. If the output symbol node is assigned to the j-th window, where $j < i$, then the input symbol node in the i-th class cannot be adjacent to that output symbol node. Otherwise, the probability that the input symbol node in the i-th class is adjacent to the 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 $\Omega^{(j)}(x)$. 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

$$p_i^{ML}(\varepsilon) \geq (1 - \sum_{j=i}^{r}\frac{\Gamma_j\mu_j}{k_j})^{k(1+\varepsilon)}. \tag{9}$$

The upper bound on the bit error rate of the input symbols from different importance classes of EWF codes is derived similarly as for LT codes in [6]. More precisely, it is the sum of probabilities that a vector $x \in \mathbf{F}_2^k$, 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, over all possible arrangements of non-zero elements in the vector x. The upper bound on 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 and under the ML decoding is given by

$$p_i^{ML}(\varepsilon) \leq \min\left\{1, \right.$$

$$\sum_{t_r=1}^{k}\cdots\sum_{t_i=1}^{t_{i+1}}\sum_{t_{i-1}=0}^{t_i-1}\cdots\sum_{t_1=0}^{t_2}\prod_{p=1}^{r}\binom{k_p-k_{p-1}-\delta(p-i)}{t_p-t_{p-1}-\delta(p-i)}$$

$$\left.\cdot\left(\sum_{j=1}^{r}\Gamma_j\sum_{d=1}^{k_j}\Omega_d^{(j)}\quad\frac{\sum_{s=0}^{\lfloor d/2\rfloor}\binom{t_j}{2s}\binom{k_j-t_j}{d-2s}}{\binom{k_j}{d}}\right)^{(1+\varepsilon)k}\right\}. \quad (10)$$

Figure 3 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 (8). 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 decrease the MIB bound, while preserving the LIB bound 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 figure 3. This illustrates how EWF codes may be improved by adapting distribution $\Omega^{(1)}$.
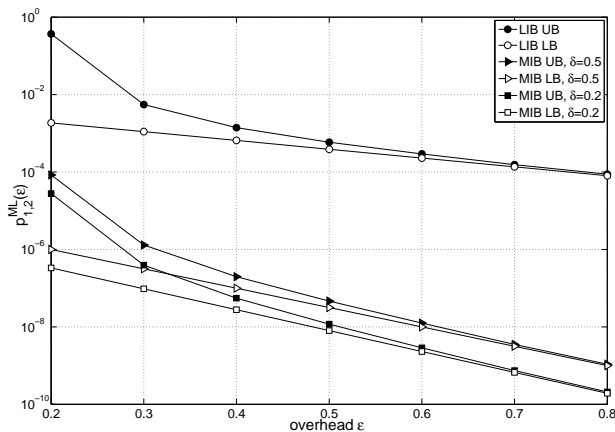

Fig. 3. Upper and lower bounds for the ML decoding of EWF codes

V. SIMULATION RESULTS

In order to verify the results of the developed asymptotic and-or tree analysis, we performed simulations 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 Figure 2 and at the receiver side, standard iterative message-passing decoding algorithm was used. Figure 4 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.
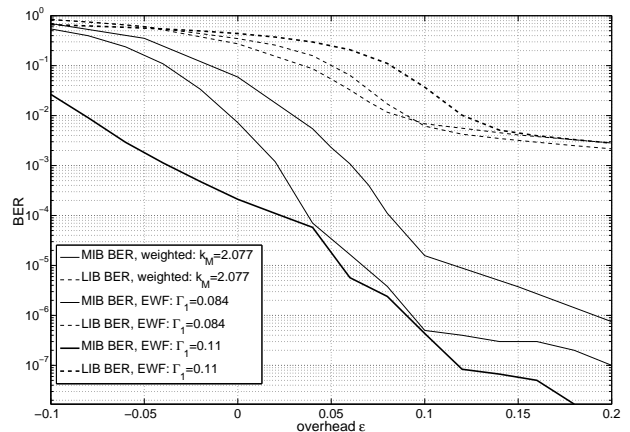

Fig. 4. Simulation results for $k = 5000$

VI. CONCLUSION

Coding applications such as reliable transmission of video files compressed with a layered coder benefit from coding schemes which offer better error protection to a certain, predefined portion of the file. Since fountain codes are an attractive solution for multicast transmission of such files, it is worthwhile to consider fountain coding techniques which offer UEP and URT properties. In this paper, we present an alternative way to construct such UEP fountain codes, by utilizing the idea of "windowing" the data set. Both analytical techniques and extensive simulations are used to show 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.

REFERENCES

[1] J. Byers, M. Luby, M. Mitzenmacher and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *Proc. of the IEEE SIG-COMM*, 1998.
[2] M. Luby, "LT Codes," *Proc. of the 43rd Annual IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, Canada, November 2002.
[3] A. Shokrollahi, "Raptor Codes," *IEEE Trans. on Information 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, December 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. on Information Theory*, vol. 53, No. 4, pp. 1521–1532, April 2007.
[7] C. Studholme and I. Blake, "Windowed Erasure Codes," *Proc. of the ISIT 2006*, pp. 509–513, Seattle, USA, July 2006.
[8] M. C. O. Bogino, P. Cataldi, M. Grangetto, E. Magli and G. Olmo, "Sliding-Window Digital Fountain Codes for Streaming Multimedia Contents" *to appear, Proc. of the ICASP 2007*, New Orleans, USA, May 2007.
[9] M. Luby, M. Mitzenmacher and A. Shokrollahi, "Analysis of random processes via and-or tree evaluation" *Proc. of the 9th Annual SIAM Symp. on Discrete Algorithms*, pp. 364–373, 1998.