Design of Unequally Error Protecting Low-Density Random Linear Fountain Codes

Birgit Schotsch and Peter Vary
Institute of Communication Systems and Data Processing (ind)
RWTH Aachen University, Aachen, Germany
{schotsch|vary}@ind.rwth-aachen.de

***Abstract*—Low-density random linear fountain (LDRLF) codes are a type of LT (Luby transform) codes with optimum erasure correction under maximum likelihood (ML) decoding given a certain density or average check node degree. The upper bound on the residual symbol erasure rate is very tight and can be used for the design of LDRLF codes instead of performing time-consuming simulations. Using LDRLF codes for unequal error protection (UEP), the excellent erasure correction performance is maintained as well as the tightness of the upper bounds for each importance class. Since the UEP upper bounds may be complex to compute, we provide an extremely good approximation thereof which is well suited to design UEP LDRLF codes. Furthermore, we provide a heuristic criterion that has to be fulfilled in order to yield good approximations.**

I. Introduction

Fountain codes are rateless erasure correcting codes that have been proposed in [1] as an alternative to retransmission schemes to recover packets in lossy packet-switched communication networks. LT (Luby transform) codes [2], Raptor codes [3] and Online codes [4] are practical realisations of rateless codes for which simple and efficient encoding and decoding algorithms exist. Originally, rateless codes have been developed for the binary erasure channel (BEC) where they exhibit the universality property, i.e. these codes perform close to the optimum at any erasure probability smaller than one. Especially in point-to-multipoint transmission scenarios, where the users experience individual and independent packet losses, this property is very useful.

The first works on rateless codes aimed at equal error protection (EEP) of data, which is used, e.g. for the distribution of bulk data, while in the recent years some proposals for rateless codes with unequal error protection (UEP) have followed to support, e.g. audio or video transmission where some parts of the data are more important than others and therefore need a stronger protection. The first rateless UEP scheme for LT codes is the weighted UEP approach by Rahnavard et al. [5], followed by the expanding window (EW) method by Sejdinovic et al. [6]. In the following, we consider only the weighted UEP method, however, with a modified code construction based on *biased sampling* [7]. In contrast to the original weighted UEP method in [5], where only discrete protection levels can be attained, biased sampling of the input symbols allows for *continuous* effective weights of the differently important data parts which implies precisely adjustable protection levels.

This work has been supported by the UMIC Research Centre, RWTH Aachen University.

Our work aims at codes for delay sensitive applications, and thus only short codes are considered. Since for short codes maximum likelihood (ML) decoding is the only decoding algorithm that shows a good error correcting performance, it is preferred to belief propagation (BP) decoding, despite the higher complexity of the former. Furthermore, we focus on so-called low-density random linear fountain (LDRLF) codes [8], [9], a type of LT codes that exhibits optimal erasure correction performance under ML decoding given a fixed average degree or density. Both the erasure correction performance as well as the computational complexity increase with the density [9]. Therefore, the density should be minimised as far as possible, provided that the target erasure correction performance is achieved.

In [7] upper and lower bounds on the residual symbol erasure probability have been derived for UEP LT codes under ML decoding for the biased sampling construction method. However, the computation of these bounds (especially the upper bounds) (see [7, Eqs. (5) and (6)]) can be quite time-consuming if the input size is large ($\gg 100$), since the complexity is polynomial in the input size. An additional significant slowdown occurs if there are more than 3 or 4 importance classes or if the degree distribution is not sparse, i.e. if the number of degrees with non-zero probability is large. The computation of the bounds is one of the first steps in the design of LT codes, but for UEP LT codes this step can become unhandy.

For (UEP) LDRLF codes the upper bound is extremely close to the actual symbol erasure rate after ML decoding [8], [9]. Thus, the upper bound can be used for the design of appropriate codes instead of running time-consuming simulations. However, the degree distribution of LDRLF codes is not sparse which is one of the main factors that increase the computation time of the upper bound and thus diminishes the time savings of code design by bound computation instead of by running simulations. Nevertheless, we have found a way to design UEP LDRLF codes by using the upper bounds of equivalent EEP LDRLF codes that can be computed very fast.

The paper is organised as follows: in Section II, we give a brief overview on LT codes over higher order Galois fields and on LDRLF codes. We summarise the original weighted UEP approach in Section III and explain the modification introduced by using biased sampling in Section III-A. The main part, i.e. the design of UEP LDRLF codes by means of the upper bounds of equivalent EEP LDRLF codes is detailed in Section IV.

II. LT Codes over $\mathbb{F}_q$

We consider LT codes over Galois fields $\mathbb{F}_q$ of order $q = 2^m$, where $m \geq 1$. With rateless codes, the transmitter can generate a potentially infinite number n_T of encoded symbols $\mathbf{y} = (y_1, y_2, \ldots y_{n_\mathrm{T}})$ from a finite amount of k_q input symbols $\mathbf{u} = \left(u_1, u_2, \ldots u_{k_q}\right)$. Though in practice the input and output symbols u_i and y_j consist of l $\mathbb{F}_q$-elements each, we use $l = 1$ in the following, since the erasure correction performance of the codes is independent of l [2]. $\mathbb{F}_q$-elements have an equivalent binary representation requiring m bits per element. To allow for a fair comparison of codes over Galois fields of different orders, we fix the number $k = k_2$ of input *bits* and distribute them to $k_q = \lceil \frac{k_2}{\mathrm{ld}\, q} \rceil = \lceil \frac{k_2}{m} \rceil$ input *symbols*. The input size of a code over $\mathbb{F}_q$ with $q = 2^m$ is thus k_q. Using good rateless codes, the receiver is able to decode the original k_q input symbols $\mathbf{u}$ from any $n_\mathrm{R} = k_q(1 + \varepsilon_\mathrm{R})$ received code symbols with high probability if the relative reception overhead is $\varepsilon_\mathrm{R} \geq 0$ but small.

The generator matrix $\mathbf{G} \in \mathbb{F}_q^{n_\mathrm{T} \times k_q}$ of an LT code, with $q = 2^m$, defines a weighted graph that connects the set of k_q input nodes $\mathbf{u} \in \mathbb{F}_q^{1 \times k_q}$ to the set of n_T output nodes $\mathbf{y} \in \mathbb{F}_q^{1 \times n_\mathrm{T}}$, where n_T can be arbitrarily large. A more detailed description of *binary* LT codes can be found in [2].

The input symbols are assigned to input nodes and the output symbols are assigned to output nodes that are also called check nodes. In vector-matrix notation we encode by $\mathbf{y}^\mathrm{T} = \mathbf{G}\mathbf{u}^\mathrm{T}$. In contrast to traditional block codes, $\mathbf{G}$ is generated online and may differ for each data block. $\mathbf{G}$ is assumed to be known at the decoder. This can be achieved by synchronised pseudo-random processes that produce $\mathbf{G}$.

The erasure correcting performance of an LT code is largely defined by its check node degree distribution $\Omega_1, \Omega_2, \ldots \Omega_{k_q}$, where a check node has degree $d \in \{1, 2, \ldots k_q\}$, i.e. it is connected to d distinct input nodes, with probability Ω_d. The degree distribution is often described by its generating polynomial $\Omega(x) = \sum_{d=1}^{k_q} \Omega_d\, x^d$. For EEP the d connected input nodes are chosen uniformly at random, i.e. with probability $p = \frac{1}{k_q}$, from the set of k_q input nodes, while for UEP the input nodes are first assigned to T importance classes. The input nodes of different classes have different probabilities of being chosen to be connected to a check node. The exact UEP construction is explained in Section III.

The d non-zero entries in a row of the generator matrix $\mathbf{G}$ correspond to the weights of the d edges between a check node and d input nodes. The value of a check node is determined by adding up the product of each value of the d input nodes with the weight of the corresponding connecting edge. The non-zero entries of $\mathbf{G}$ are sampled uniformly at random from the set of $q - 1$ non-zero $\mathbb{F}_q$-elements.

At the encoder, n_T output symbols are generated, which are then transmitted over a symbol erasure channel (SEC) that randomly erases some of the transmitted Galois field symbols. Finally, the decoder tries to reproduce the original k_q input symbols from the $n_\mathrm{R} \leq n_\mathrm{T}$ received symbols. Having collected n_R output symbols, the decoder uses the n_R rows of $\mathbf{G}$ that are associated with the collected, non-erased symbols to form a new matrix $\mathbf{G}'$ which is used for decoding. Since $\mathbf{G}'$ consists of a set of n_R rows sampled at random from the original matrix $\mathbf{G}$ according to the erasures that occur on the SEC, $\mathbf{G}'$ has the same statistical properties as $\mathbf{G}$.

The most popular decoding algorithm for LT codes is the computationally cheap, though suboptimal, BP decoding algorithm that performs well on properly designed codes, but only for large blocklengths. The optimal decoding algorithm (optimal in the sense of minimal symbol erasure probability at a certain reception overhead) is ML decoding, which, in the case of the erasure channel, is equivalent to solving a consistent system of n_R linear equations in k_q unknowns by means of Gaussian elimination (GE). GE is computationally expensive for large input sizes, but it is affordable for short codes. Moreover, considering delay sensitive applications, the usage of short codes is inevitable and in such a case the ML decoding algorithm is the only decoding algorithm that exhibits a good performance.

A. Low-Density Random Linear Fountain Codes

Low-density random linear fountain (LDRLF) codes [8], [9] are a class of LT codes with optimal erasure correction performance under ML decoding, given a fixed density Δ or average degree $\bar{\Omega} = \Delta \cdot k_q = \sum_{d=1}^{k_q} d\, \Omega_d$ and a fixed input size k_q. The density Δ of an LT code over $\mathbb{F}_q$ is the ratio of the number of non-zero entries in the generator matrix to the total number of entries.

LDRLF codes have the following degree distribution[1]:

$$\begin{aligned} \Omega(x) &= \sum_{d=0}^{k_q} \Omega_d x^d \\ &= \sum_{d=0}^{k_q} \binom{k_q}{d} P_{\neg 0}^{d} (1 - P_{\neg 0})^{(k_q - d)} x^d \qquad (1) \end{aligned}$$

which results from sampling a zero entry in the generator matrix $\mathbf{G}$ with probability $P_0 > \frac{1}{q}$ and a non-zero entry with probability $P_{\neg 0} < \frac{q-1}{q}$. The non-zero entries are chosen uniformly at random from the set of $q - 1$ non-zero Galois field elements. The probability $P_{\neg 0}$ is equivalent to the density Δ. In contrast to conventional random linear fountain (RLF) codes [3], [10], [11], the low-density variant is encodable and decodable with a significantly lower complexity.

For EEP LDRLF codes a lower and an upper bound on the symbol erasure probability $P_q^{[\mathrm{ML,S}]}$ are given in [9]. The lower

[1] In practical systems and also in our simulations Ω_0 equals zero. Thus, a modification of the probabilities Ω_d for $d > 0$ is necessary, in order to obtain $\sum_{d=1}^{k_q} \Omega_d = 1$ and to maintain a constant average check node degree. However, for not too small input sizes k_q or average check node degrees $\bar{\Omega}$, the induced error of considering $\Omega_0 \neq 0$ as in (1) is negligible.

bound is

$$\underline{P}_q^{[\mathrm{ML,S}]} = \left(1 - \frac{\bar{\Omega}}{k_q}\right)^{k_q \gamma_{\mathrm{R}}} = (1 - P_{\neg 0})^{k_q \gamma_{\mathrm{R}}} \tag{2}$$

and the upper bound is

$$\overline{P}_q^{[\mathrm{ML,S}]} = \sum_{w=1}^{k_q} \binom{k_q - 1}{w - 1} (q-1)^{w-1} \cdot \left(\frac{1}{q} + \frac{q-1}{q} \left(1 - \frac{q}{q-1} P_{\neg 0}\right)^{w} \right)^{k_q \gamma_{\mathrm{R}}}, \tag{3}$$

where $\gamma_{\mathrm{R}} = 1 + \varepsilon_{\mathrm{R}}$ is the inverse reception rate. For notational convenience, we will implicitly assume that probabilities and their bounds are limited from above by one, i.e. the operation $\min\{1, \cdot\}$ is omitted. An important feature of the above upper bound of LDRLF codes is that it is extremely close to the actual symbol erasure probability $P_q^{[\mathrm{ML,S}]}$, i.e. it can be used for code design instead of time-consuming simulations. This characteristic feature can also be found with the upper bound on the symbol erasure rate of each importance class when constructing UEP LDRLF codes.

III. Weighted Unequal Error Protection

The k_q input nodes are first assigned to T importance classes, where importance class τ has size $k_{q,\tau} = \alpha_\tau k_q$, with $1 \leq \tau \leq T$, $0 \leq \alpha_\tau \leq 1$ and $\sum_{i=1}^{T} \alpha_i = 1$, where α_τ is the relative size of class τ. According to the importance of the classes, weighting factors ω_τ are chosen such that the new initial probability of connecting input node ι from class τ to the current check node is $p_{\tau,\iota} = \frac{\omega_\tau}{k_q} = \omega_\tau p$. Since $p_{\tau,\iota}$ is equal for all input nodes ι within class τ we simply use p_τ instead of $p_{\tau,\iota}$. Thus, $\sum_{i=1}^{T} p_i k_{q,i} = \sum_{i=1}^{T} \omega_i \alpha_i = 1$. In the original weighted UEP approach [5, Sec. IV] the number d_τ of input nodes from an arbitrary class τ that are connected to a check node of degree d is set to $\min([\alpha_\tau \omega_\tau d], k_{q,\tau})$, where $[x]$ means rounding to the nearest integer. Although this rounding operation simplifies the analysis significantly, it comes with a major drawback, namely it causes the effective weights

$$\omega_\tau^{[\mathrm{eff}]} = \frac{\bar{\Omega}_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d=1}^{d_{\max}} \Omega_d \min([\alpha_\tau \omega_\tau d], k_{q,\tau})}{\alpha_\tau \bar{\Omega}}, \tag{4}$$

as given in [7] to deviate from the target weights ω_τ. The numerator in (4) is the effective average degree $\bar{\Omega}_\tau$ of class τ, while the denominator is the average degree of class τ in the EEP case. Furthermore, only a discrete set of effective weights $\omega_\tau^{[\mathrm{eff}]}$ is obtained. This means that the effective protection levels deviate from the targeted ones and also the set of protection levels becomes discrete.

A. UEP LT Code Construction by Biased Sampling

In order to allow for a continuous relation between the target protection levels ω_τ and the effective ones $\omega_\tau^{[\mathrm{eff}]}$, a different construction method has been analysed in [7], where biased sampling is used to select the input nodes from the different

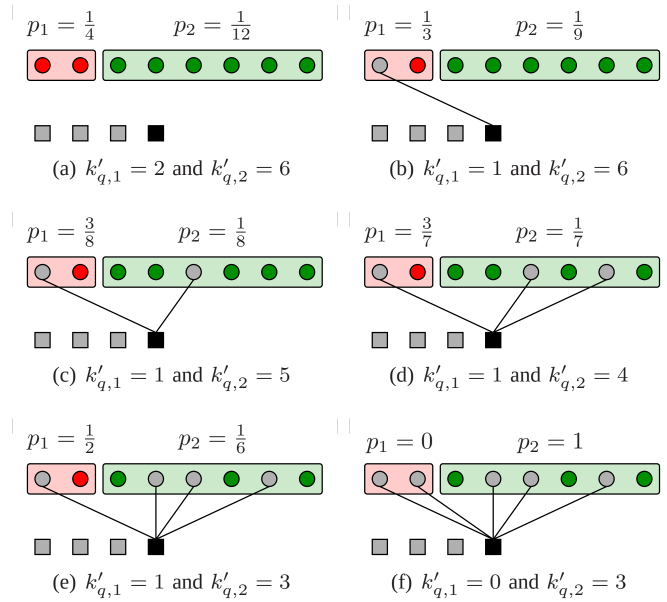


Fig. 1. One possible realisation of connecting source nodes to a check node of degree $d = 5$ via biased sampling given an input size of $k_q = 8$, $T = 2$ importance classes of relative sizes $\alpha_1 = 0.25$ and $\alpha_2 = 0.75$ and class weights $\omega_1 = 2$ and $\omega_2 = \frac{2}{3}$. After each draw, the probabilities p_1 and p_2 of connecting an edge to a specific input node of class 1 or 2 have to be updated, taking into account the remaining $k'_{q,1}$ and $k'_{q,2}$ unconnected input nodes in either class, i.e. $p_1 = \frac{\omega_1}{\omega_1 k'_{q,1} + \omega_2 k'_{q,2}}$ and $p_2 = \frac{\omega_2}{\omega_1 k'_{q,1} + \omega_2 k'_{q,2}}$.

classes to be connected to the current check node of degree d. We review the relevant parts of this approach using the same notation as in [7]. Additionally, we illustrate the sampling process by means of a comprehensive example.

Biased sampling of input nodes can be described by an equivalent urn model, where d balls are drawn one by one without replacement from an urn that contains $k_q = \sum_{i=1}^{T} k_{q,i}$ balls of T different colours and each ball of colour τ has weight ω_τ. The probability of picking a ball of a particular colour at a particular draw is proportional to its relative weight with respect to the total weight of the remaining balls. An example is provided in Fig.1. Biased sampling has been analysed by Wallenius [12] for the univariate case ($T = 2$). The generalisation to the multivariate case is due to Chesson [13]. The partitioning of the overall degree d into *class degrees* d_τ, with $\sum_{\tau=1}^{T} d_\tau = d$, consequently can be characterised by the so-called multivariate Wallenius' noncentral hypergeometric distribution [12], [13], which expresses the conditional probability mass function (pmf)

$$P\left(d_1, \dots d_{T-1} \middle| d; \mathbf{k_q}, \boldsymbol{\omega}\right) = \left(\prod_{i=1}^{T} \binom{k_{q,i}}{d_i}\right) \int_0^1 \prod_{i=1}^{T} \left(1 - t^{\frac{\omega_i}{\boldsymbol{\omega}(\mathbf{k_q} - \mathbf{d})}}\right)^{d_i} dt. \tag{5}$$

The vectors $\mathbf{d}$, $\mathbf{k_q}$ and $\boldsymbol{\omega}$ denote the class degrees $\mathbf{d} = (d_1, d_2, \dots d_T)$, the class sizes $\mathbf{k_q} = (k_{q,1}, k_{q,2}, \dots k_{q,T})$ and class specific target weights $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots \omega_T)$. Eq. (5) can be evaluated by numerical integration as described in [14] using the BiasedUrn R package [15]. Without loss of generality, the degree d_T is not explicitly mentioned in the left-hand side of (5), since it is included implicitly due to $\sum_{\tau=1}^{T} d_\tau = d$. We omit the parametrisation with the class sizes $\mathbf{k_q}$ and weights $\boldsymbol{\omega}$ to simplify the notation and define the joint pmf as

$$\begin{aligned} P(\mathbf{d}) &= P(d_1, \ldots d_T) = P(d_1, \ldots d_{T-1}, d) \\ &= P(d) \cdot P\left(d_1, \ldots d_{T-1} \middle| d; \mathbf{k_q}, \boldsymbol{\omega}\right), \end{aligned} \tag{6}$$

where $P(d) = \Omega_d$, i.e. the coefficients Ω_d of the check node degree distribution $\Omega(x)$.

In the remainder of this paper we will use the following simplified notation: Given an arbitrary function $f(\mathbf{d})$, the collated sum

$$\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} f(\mathbf{d}) \qquad \text{denotes} \qquad \sum_{d_1} \cdots \sum_{d_T} f(\mathbf{d}),$$

where the sums are calculated for all combinations of the values of $\mathbf{d} = (d_1, d_2, \ldots d_T)$ for which $1 \leq d \leq d_{\max}$ and $\sum_{\tau=1}^{T} d_\tau = d$. Additionally, $0 \leq d_\tau \leq \min(d, k_{q,\tau})$ with $1 \leq \tau \leq T$.

In the case of the biased sampling code construction method, the effective weight is

$$\omega_\tau^{[\text{eff}]} = \frac{\bar{\Omega}_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d_\tau} P(d_\tau)\, d_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d})\, d_\tau}{\alpha_\tau \bar{\Omega}},$$

where $\bar{\Omega}_\tau$ is the average degree of class τ and $P(d_\tau)$ is obtained by marginalising $P(\mathbf{d})$.

A lower and an upper bound on the symbol erasure rate $P_{q,\tau}^{[\text{ML,S}]}$ of importance class τ are given by [7]

$$\underline{P}_{q,\tau}^{[\text{ML,S}]} = \left(1 - \sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d}) \frac{d_\tau}{k_{q,\tau}}\right)^{k_q \gamma_{\text{R}}} \tag{7}$$

and by eq. (8) on the next page, respectively.

IV. Approximation of Performance Bounds

The computation of these bounds (especially the upper bound) can be quite time-consuming if the input size is large ($k_q \gg 100$), if the number τ of importance classes is greater than 3 or 4 or if the degree distribution is not sparse. However, the approximation of the UEP code bounds with the bounds of equivalent EEP codes is only feasible for LDRLF codes, not for LT codes in general. Since LDRLF codes are parametrised with the average degree $\bar{\Omega}$ (or the density Δ) and the input size k_q, equivalent EEP LDRLF codes can be found easily: an EEP LDRLF code is suited to approximate the bounds of importance class τ of a UEP LDRLF code if the codes have the same overall input size k_q and if the EEP generator matrix $\mathbf{G}_\tau^{[\text{EEP}]}$ has the same density as the UEP generator matrix $\mathbf{G}^{[\text{UEP}]}$ in the columns that cover importance class τ, i.e. $\Delta_\tau = \Delta_\tau^{[\text{EEP}]} = \Delta_\tau^{[\text{UEP}]} = \bar{\Omega}_\tau / k_{q,\tau}$. The design of an equivalent EEP LDRLF code is illustrated in Fig. 2.

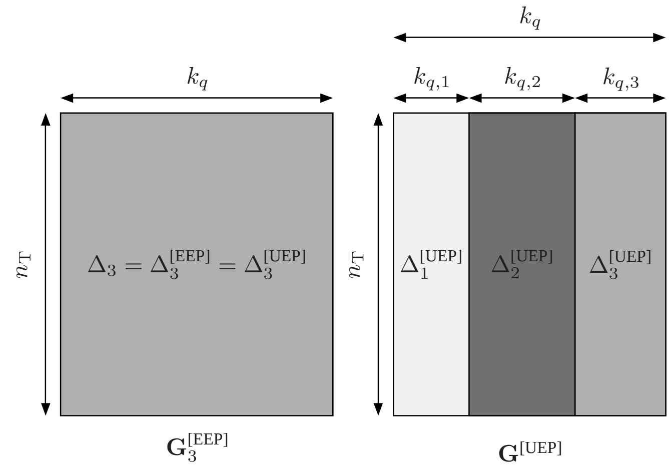


Fig. 2. Illustration of the design of an EEP LDRLF code that can be used to approximate the bounds of the third importance class of a UEP LDRLF code which is constructed by means of biased sampling.

In Fig. 4 upper and lower bounds on the symbol erasure probability $P_{q,\tau}^{[\text{ML,S}]}$ under ML decoding are depicted in red for three different binary UEP LT codes A, B and C that are constructed by biased sampling. The input size is $k_2 = 100$ and there are $T = 2$ importance classes of relative sizes $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. For example by using the red characteristics in Fig. 3 for the respective codes, the weights ω_τ can be determined such that the effective weights are $\omega_1^{[\text{eff}]} = 1.5$ and $\omega_2^{[\text{eff}]} = 0.9444$.

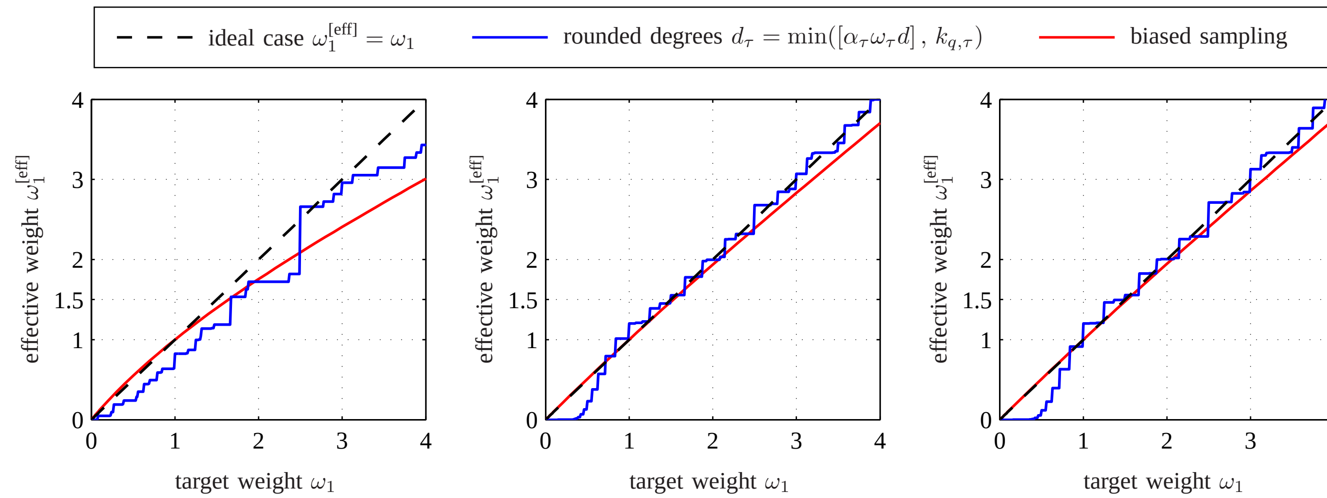
Code A is a UEP LT code that is based on the well-known degree distribution $\Omega_{\text{A}}(x) = 0.007969x + 0.49357x^2 + 0.16622x^3 + 0.072646x^4 + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 + 0.05559x^{19} + 0.025023x^{65} + 0.003135x^{66}$ which is taken from [3] and that has been optimised for belief propagation (BP) decoding. The overall average degree is $\bar{\Omega}_{\text{A}} = 5.87$, while the average degree of class τ is $\bar{\Omega}_{\text{A},\tau} = \omega_\tau^{[\text{eff}]} \alpha_\tau \bar{\Omega}_{\text{A}}$, i.e. $\bar{\Omega}_{\text{A},1} = 0.8805$ and $\bar{\Omega}_{\text{A},2} = 4.9895$. Code C is a UEP LDRLF code that is designed to have the same average degrees as code A. Code A is just included to show the difference in the erasure correction performance of a BP optimal code and an ML optimal code of the same density. Code B is a UEP LDRLF code with an overall average degree $\bar{\Omega}_{\text{B}} = 7$. The class specific average degrees are $\bar{\Omega}_{\text{B},1} = 1.05$ and $\bar{\Omega}_{\text{B},2} = 5.95$.

For the LDRLF codes (codes B and C), the upper bounds are approximated by the bounds of equivalent EEP codes which are depicted in black in Fig. 4. The lower bounds of EEP and UEP codes are equal (no approximation), while the approximations to the upper bounds of code B (the denser code) match significantly better than the ones for code C.

For a sufficiently accurate approximation, the following condition has to be fulfilled. Without loss of generality let class T be the least protected class. Then the average degree $\bar{\Omega}_T$ of the least protected class should be

$$\bar{\Omega}_T \geq c \cdot \alpha_T, \tag{9}$$

$$\overline{P}_{q,\tau}^{[\mathrm{ML,S}]} = \sum_{\substack{w=w_1+\ldots+w_T=1 \\ w_\tau \geq 1}}^{k_q} (q-1)^{w-1} \left(\prod_{i=1}^{T} \binom{k_{q,i}-\delta_{\tau-i}}{w_i-\delta_{\tau-i}} \right) \cdot \left(\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d}) \sum_{s=s_1+\ldots s_T=0}^{d} \frac{1}{q} \left(1-(1-q)^{1-s}\right) \prod_{i=1}^{T} \frac{\binom{w_i}{s_i}\binom{k_{q,i}-w_i}{d_i-s_i}}{\binom{k_{q,i}}{d_i}} \right)^{k_q \gamma_{\mathrm{R}}} \tag{8}$$



(a) UEP LT code with $\Omega_{\mathrm{A}}(x)$.

(b) UEP LDRLF code with $\Omega_{\mathrm{B}}(x)$.

(c) UEP LDRLF code with $\Omega_{\mathrm{C}}(x)$.

Fig. 3. The effective weight $\omega_1^{[\mathrm{eff}]}$ of class 1 as a function of the target weight ω_1 for three UEP LT codes of input size $k_2 = 100$ and two classes of relative sizes $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. Since the red curves (biased sampling) are continuous, the deviations from the ideal line can be compensated easily.

where c is at least 6. This heuristic arises from the observation that EEP LDRLF codes with too low average degrees ($\bar{\Omega} < 6$) degenerate, i.e. the characteristic waterfall region is very small and converges directly to the high error floor. The error floor lowers with increasing average degree. The effect of a too low average degree in importance class 2 can be observed for code C in Fig. 4(c). With $\bar{\Omega}_{\mathrm{C},2} = 4.9895 < 6 \cdot 0.9 = 5.4$, the above condition is not fulfilled by code C. Since the two classes are connected by common check nodes, the less protected class weakens the better protected class and the better protected class improves decoding for the less protected class, both compared to the corresponding EEP code. Especially the weakening of the better protected class is clearly visible as this class is very small compared to the less protected class.

If condition (9) is met, which is the case for code B, where we have $\bar{\Omega}_{\mathrm{B},2} = 5.95 > 5.4$, the UEP upper bounds and their EEP approximations virtually coincide. In Fig. 4(b) they overlap for both importance classes.

A design rule for UEP LT codes in general is to set the weights such that the resulting density of the best protected class does not exceed $1 - \frac{1}{q}$. Increasing the density beyond $1 - \frac{1}{q}$ would decrease the protection level again.

V. Conclusion

Low-density random linear fountain (LDRLF) codes exhibit their excellent erasure correction performance under maximum likelihood (ML) decoding also when used for unequal error protection (UEP). The tightness of the upper bound on the symbol erasure probability in the case of equal error protection (EEP) makes this bound an ideal tool to design LDRLF codes without extensive simulations. Using LDRLF codes for UEP this tightness of the upper bounds is preserved for all importance classes. Although the computation of the UEP upper bounds (8) can be quite complex, we have shown that using the upper bounds (3) of equivalent EEP LDRLF codes constitutes a simple and highly accurate alternative to the computation of the UEP bounds.

References

[1] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 56–67, 1998.

[2] M. Luby, "LT Codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

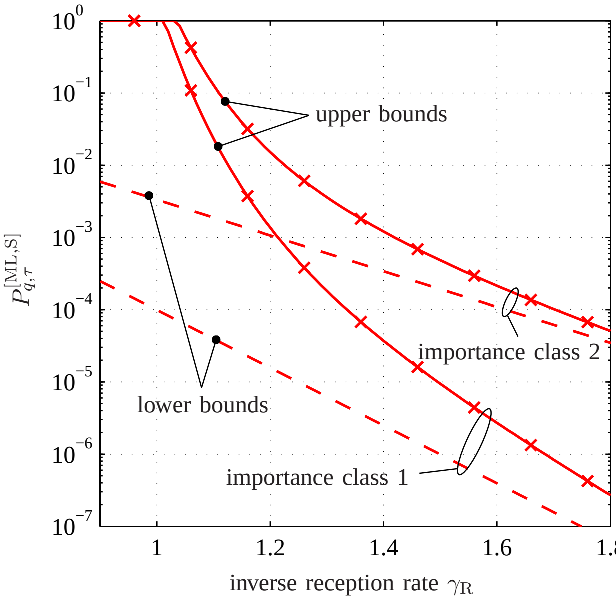
[3] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[4] P. Maymounkov, "Online Codes," Secure Computer Systems Group, New York University, Tech. Rep. TR2002-833, Nov. 2002.

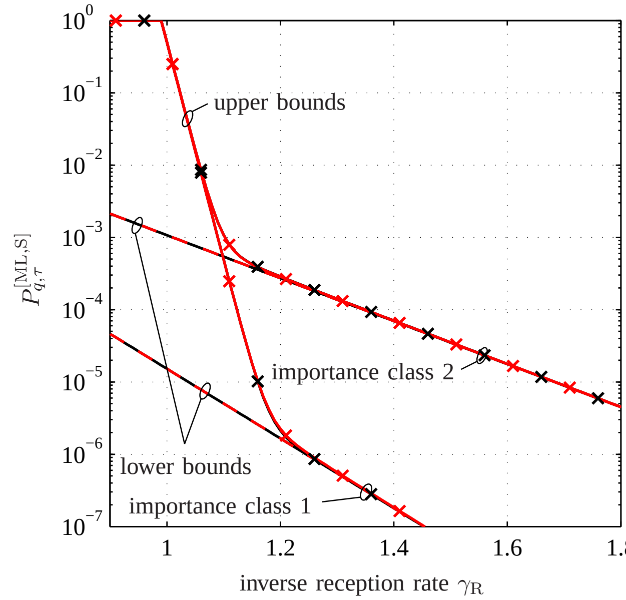
[5] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless Codes With Unequal Error Protection Property," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[6] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding Window Fountain Codes for Unequal Error Protection," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.

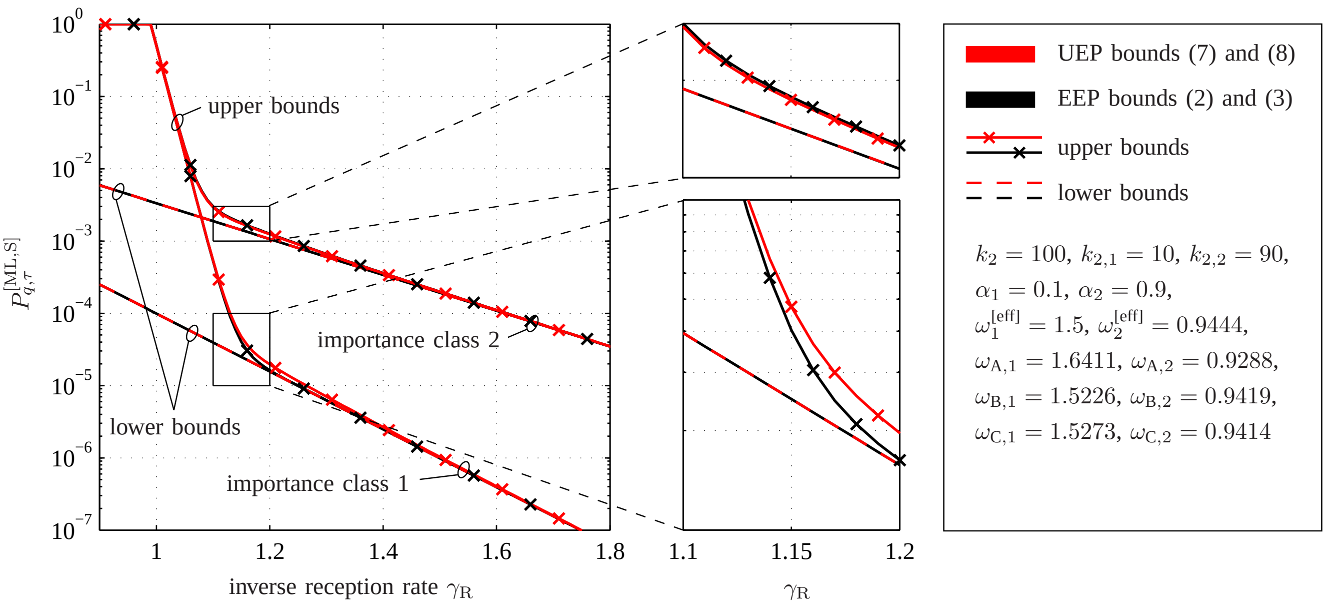
[7] B. Schotsch and R. Lupoaie, "Finite Length LT Codes over $\mathbb{F}_q$ for Unequal Error Protection with Biased Sampling of Input Nodes," in *IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, USA, July 2012.



(a) $\Omega_{\mathrm{A}}(x)$, $\Delta_{\mathrm{A}} = 5.87\%$, $\Delta_{\mathrm{A},1} = 8.81\%$, $\Delta_{\mathrm{A},2} = 5.54\%$.



(b) LDRLF code, $\Omega_{\mathrm{B}}(x)$, $\Delta_{\mathrm{B}} = 7.00\%$ $\Delta_{\mathrm{B},1} = 10.5\%$, $\Delta_{\mathrm{B},2} = 7.78\%$.



(c) LDRLF code, $\Omega_{\mathrm{C}}(x)$, $\Delta_{\mathrm{C}} = 5.87\%$, $\Delta_{\mathrm{C},1} = 8.81\%$, $\Delta_{\mathrm{C},2} = 5.54\%$.

Fig. 4. Upper and lower bounds on the symbol erasure rate $P_{q,\tau}^{[\mathrm{ML,S}]}$ for three binary UEP LT codes as well as the bounds of the EEP LDRLF codes that correspond to the UEP LDRLF codes.

[8] B. Schotsch, H. Schepker, and P. Vary, "The Performance of Short Random Linear Fountain Codes under Maximum Likelihood Decoding," in *IEEE International Conference on Communications (ICC)*, June 2011.

[9] B. Schotsch, R. Lupoaie, and P. Vary, "The Performance of Low-Density Random Linear Fountain Codes over Higher Order Galois Fields under Maximum Likelihood Decoding," in *49th Allerton Conference on Communication, Control and Computing*, September 2011.

[10] D. MacKay, "Fountain Codes," *Communications, IEE Proceedings-*, vol. 152, no. 6, pp. 1062–1068, 2005.

[11] G. Liva, E. Paolini, and M. Chiani, "Performance versus overhead for fountain codes over $\mathbb{F}_q$," *IEEE Communications Letters*, vol. 14, no. 2, pp. 178–180, 2010.

[12] K. T. Wallenius, "Biased sampling: The noncentral hypergeometric probability distribution," Ph.D. dissertation, Stanford University, 1963.

[13] J. Chesson, "A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation," *Journal of Applied Probability*, vol. 13, no. 4, pp. 795–797, 1976.

[14] A. Fog, "Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution," *Communications in Statistics - Simulation and Computation*, vol. 37, no. 2, pp. 258–273, 2008.

[15] ——, "BiasedUrn: Biased Urn Model Distributions," 2011, Version 1.04. [Online]. Available: http://cran.r-project.org/web/packages/BiasedUrn/index.html