UEP LT Codes with Intermediate Feedback
Jesper H. Sørensen, Petar Popovski, and Jan Østergaard

Abstract—We analyze a class of rateless codes, called Luby transform (LT) codes with unequal error protection (UEP). We show that while these codes successfully provide UEP, there is a significant price in terms of redundancy in the lower prioritized segments. We propose a modification with a single intermediate feedback message. Our analysis shows a dramatic improvement on the decoding performance of the lower prioritized segment.

Index Terms—

I. INTRODUCTION

Rateless codes are capacity achieving erasure correcting codes, which can generate a potentially infinite amount of encoded symbols from \( k \) input symbols. Decoding is possible when \((1 + \epsilon)k\), \( \epsilon \geq 0 \), encoded symbols have been received. Regardless of the channel conditions, a rateless code will approach the channel capacity, without the need for feedback. Successful examples are LT codes [1] and Raptor codes [2].

Conventional rateless codes treat all data as equally important. In some applications, e.g., video streaming [3], this is not desirable. Several works address the problem of designing rateless codes with unequal error protection (UEP), where different data segments have different error probabilities. Variants based on LT codes are found in [4]–[6], while [7] is an example using Raptor codes. Common for these is the idea of biasing the random sampling towards more important data.

In this work, analysis is presented for the UEP LT code from [6]. It acts as an analytical extension of the initial simulation results in [8], thus corroborating the conclusions therein. The purpose is to quantify the redundancy of these codes, when recovery of all data segments is desired. Evaluations show that improving the decoding performance of the more important data can significantly degrade the decoding performance of the less important data. Motivated by this result, we propose and analyze a modification of these codes, where successful decoding of a data segment is reported to the transmitter through a feedback channel. Our analysis shows that it is very beneficial to add such a single intermediate feedback.

We consider point-to-point communication, where it is justified to use the feedback. However, since the intermediate feedback is limited to only a single bit per \( k \) input symbols, the scheme may find applicability in broadcast scenarios, where many low rate feedback messages come from different users. However, that problem is out of scope for this letter, as it would involve multiuser utility optimization, such as fairness.

II. LT CODES

Assume we wish to transmit a given amount of data, which is divided into \( k \) input symbols. An encoded symbol, also called an output symbol, is generated as the bitwise XOR of \( i \) input symbols, where \( i \) is found by sampling the degree distribution, \( \pi(i) \). The value \( i \) is referred to as the degree of the output symbol, and all input symbols contained in an output symbol are called neighbors of the output symbol. The encoding process can be broken down into three steps: 1) Randomly choose a degree \( i \) by sampling \( \pi(i) \). 2) Choose uniformly at random \( i \) of the \( k \) input symbols. 3) Perform bitwise XOR of the \( i \) chosen input symbols. The resulting symbol is the output symbol. This process can be iterated as many times as needed, which results in a rateless code.

Decoding performs the reverse XOR operations from the encoding process. Initially, all degree-1 output symbols are identified, which makes it possible to recover their neighboring input symbols. These are moved to a storage called the ripple. Symbols in the ripple are processed one by one, which means they are XOR-ed with all their neighbors and removed from the ripple. This will potentially reduce some of the encoded symbols to degree one, which makes it possible for the decoder to further process symbols. When a symbol is recovered, there is a risk that it is already in the ripple and thereby redundant. This part of the relative redundancy, \( \epsilon \), is denoted \( \epsilon_R \). Decoding is successful when all input symbols have been recovered. If at any point before this, the ripple is empty, decoding has failed. The receiver then signals a failure, or waits for more symbols. In the latter case, new output symbols are initially stripped for already recovered input symbols, leaving the output symbol with a reduced degree. If the reduced degree is zero, the symbol is redundant. This part of the relative redundancy is denoted \( \epsilon_0 \), and we have that \( \epsilon = \epsilon_0 + \epsilon_R \).

We use the approach to UEP proposed in [6], where the uniform distribution used for selection of input symbols is replaced by one which favors more important symbols. Hence, in step 2 of the encoder, a non-uniform random selection of symbols is performed. This solution to UEP has no impact on the decoder. We refer to these codes as UEP LT codes.

III. DEFINITIONS AND NOTATION

Vectors are denoted in bold and indexed with subscripts, e.g., \( X_i \) is the \( i \)th element of \( X \). The sum of all elements is denoted \( X \) and the zero vector is denoted \( \mathbf{0} \). Random variables are denoted with upper case letters and a realization in lower case. The probability mass function of \( X \) is denoted \( f_X(x) \).

In UEP LT codes, the \( k \) input symbols are divided into \( N \) subsets \( s_1, s_2, \ldots, s_N \), each of size \( \alpha_1 k, \alpha_2 k, \ldots, \alpha_N k \), where \( \sum_{j=1}^{N} \alpha_j = 1 \). We refer to these subsets as layers.
We define the vector \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N] \). The probability of selecting input symbols from \( s_j \) is \( p_j(k)\alpha_jk \), such that \( \sum_{j=1}^N p_j(k)\alpha_jk = 1 \) and without loss of generality we assume that \( p_i(k) \geq p_j(k) \) if \( i < j \). Note that if \( p_j(k) = \frac{1}{k} \ \forall j \), then all data are treated equally, as in the standard single layer LT code. We define the vector, \( \beta \), with \( \beta_i = \frac{p_i(k)}{p_n(k)} \).

An encoded symbol of a UEP LT code can be seen as having at most \( N \) dimensions. The \( N \)-dimensional degree is denoted \( j \), where \( j_n \) denotes the number of neighbors belonging to the \( n \)'th layer. We refer to \( \ell \) as the reduced degree, where \( \ell'_n \) denotes the reduced number of neighbors belonging to the \( n \)'th layer. Moreover, we define \( \mathbf{L} = [L_1, L_2, \ldots, L_N] \), where \( L_n \) denotes the number of unprocessed input symbols from the \( n \)'th layer. Similarly, we define \( \mathbf{R} = [R_1, R_2, \ldots, R_N] \), where \( R_n \) is the number of symbols in the ripple belonging to the \( n \)'th layer. By \( \mathcal{J}_i \), we denote the set of \( j \) which satisfy \( j_n \geq \ell'_n, \ n = 1, 2, \ldots, N \), and \( j = i \). The number of encoded symbols collected by the receiver is denoted \( \Delta \).

**Definition 1.** (Decoder State) A decoder state, \( \mathbf{D} \), is defined by the number of unprocessed symbols, \( \mathbf{L} = [L_1, L_2, \ldots, L_N] \) and the ripple, \( \mathbf{R} = [R_1, R_2, \ldots, R_N] \). Hence, \( \mathbf{D} = [\mathbf{L}, \mathbf{R}] \).

Whenever a symbol is processed, \( \hat{L} \) will decrease by one. We call this a decoding step, which can only be performed if the ripple is not empty. If the ripple is empty, decoding stops, and we are left with what we call a terminal state.

**Definition 2.** (Terminal State) A terminal state, \( \mathbf{D}^X = [\mathbf{L}^X, \mathbf{R}^X] \), is defined as a state in which \( \mathbf{R} = 0 \). The probability of ending in a particular terminal state, \( \mathbf{D}^X \), after having collected \( \Delta \) symbols, is denoted \( f_{\mathbf{D}^X}(\mathbf{D}^X|\Delta) \).

If the decoder is in a terminal state and not all \( k \) input symbols have been decoded, more symbols are needed. Once a symbol of reduced degree 1 is received, decoding is restarted. The number of symbols collected while in terminal state \( \mathbf{D}^X \) is denoted \( \Delta_{\mathbf{D}^X} \). The maximum value of \( \Delta \) is denoted \( \Delta_{\text{max}} \).

### IV. Analysis

For a partially decoded LT code, there is a probability that the reduced degree of a new symbol is zero. Such a symbol is redundant and thus discarded. We present the reduced degree distribution for a UEP LT code and use it to derive the probability of redundancy. We show how the simple use of a feedback message can significantly decrease this probability.

#### A. Reduced Degree Distribution

In [8] the reduced degree distribution as a function of the number of unprocessed symbols from individual layers, \( \ell \), was derived for any \( \beta \). We iterate it in Theorem 1 for convenience with small notational modifications to fit the context.

**Theorem 1.** (\( N \)-Layer Reduced Degree Distribution [8]) In an \( N \)-layer UEP LT code using any \( \pi(i) \) and with parameters, \( \alpha \) and \( \beta \), where \( \ell_n \) symbols remain unprocessed from \( n \)'th layer,

\[
\begin{align*}
\n, n = 1, 2, \ldots, N, \text{ the reduced degree distribution, } & \pi'_n(\ell', \ell), \\
\text{is for a two-layer LT code with parameters } & \alpha = 0.5, \beta = 9 \text{ and } k = 100. \\
\end{align*}
\]

where \( \Phi \) is Wallenius’ noncentral hypergeometric distribution.

When evaluating \( \pi'_n(\ell', \ell) \) at \( \ell' = 0 \) we get an interesting quantity. At a given terminal state, \( \mathbf{D}^X \), when \( \ell_n^X \) symbols remain unprocessed from the \( n \)'th layer, \( n = 1, 2, \ldots, N \), \( \pi'_n(0, \ell^X) \) is the probability that the next received symbol is redundant. This is a key element of this analysis, since it enables us to evaluate the expected value of \( \epsilon_0 \).

In the further analysis, we will treat the case of \( N = 2 \), where we refer to the layers as base layer (\( n = 1 \)) and refinement layer (\( n = 2 \)). In Fig. 1, \( \pi'_2(0, \ell^X) \) has been plotted as a function of \( L_B^X \) and \( L_R^X \), the number of under-coded symbols from the base layer and the refinement layer, respectively. The parameters, \( \alpha_B = 0.5 \), henceforth denoted as \( \alpha, \alpha_R = 1 - \alpha, \beta = \frac{0.2}{0.8} = 9 \) and \( k = 100 \) have been chosen. An optimized degree distribution is not provided in [6], thus we have chosen the Robust Soliton distribution (RSD) from [1] with parameters \( c = 0.1 \) and \( \delta = 1 \). The plot shows that the probability of redundancy increases faster for decreasing \( L_B \) than for decreasing \( L_R \), which is expected since the base layer symbols are more likely to occur as neighbors.

#### B. UEP LT Codes without Feedback

We will now use \( \pi'_n(0, \ell^X) \) to derive the expected redundancy, \( \mathbb{E}[k\epsilon_0(\Delta_{\text{max}})] \), due to symbols having reduced degree zero. This is a significant extension of the work in [8] and is the key analytical contribution of this letter. In order to do this, we must find the expected number of symbols received in each possible state. In this regard, we note that the \( \Delta \)th symbol is received in state \( \mathbf{D}^X \), if the decoding of the first \( \Delta - 1 \) symbols resulted in the terminal state \( \mathbf{D}^X \). Hence, the expected number of symbols, \( \mathbb{E}[\Delta_{\mathbf{D}^X}(\Delta_{\text{max}})] \), received while being in state \( \mathbf{D}^X \), equals the expected number of times decoding fails in that state. From \( \mathbb{E}[\Delta_{\mathbf{D}^X}(\Delta_{\text{max}})] \), it is easy to obtain
Theorem 2. (Redundancy in Two-Layer UEP LT Code) In a two-layer UEP LT code using any degree distribution, \( \pi(i) \), and a single feedback message when the base layer has been decoded, the expected total redundancy, \( \mathbb{E}[k_0^{\ell}(\Delta_{max})] \), due to reduced degree zero, is:

\[
\mathbb{E}[k_0^{\ell}(\Delta_{max})] = \sum_{\Delta=1}^{\Delta_{max}} f_{D^X}(d^X|\Delta).
\]

Proof: The probability of ending in the terminal state \( d^X \), when trying to decode the first \( \Delta \) symbols, is given by \( f_{D^X}(d^X, \Delta) \). The expected number of symbols, \( \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \), received in this state during an entire transmission is found by summing \( f_{D^X}(d^X, \Delta) \) for all possible \( \Delta \), i.e., \( \Delta = 1, 2, \ldots, \Delta_{max} \). Of these, a fraction \( \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \pi_\beta(0, \ell^X) \) have reduced degree 0. Finally, the total amount of symbols, \( \mathbb{E}[e_k] \), with reduced degree 0 is found by summing over all terminal states in which the transmission continues, i.e., any state in which \( \ell^X \neq 0 \).

Lemma supporting Theorem 2, which enables the evaluation of \( f_{D^X}(d^X|\Delta) \) have been derived. However, for large \( k \), the complexity is prohibitive. We have therefore resorted to Monte Carlo simulations, for which examples are given in section V. Consequently, the lemmas are omitted in this letter, but can be found in an unpublished technical note [9].

C. UEP LT Codes with Feedback

We now treat the case where an intermediate feedback message is applied. It informs the transmitter that the base layer has been decoded. The transmitter adapts by excluding the base layer symbols from the random selection in step 2 of the encoder. The feedback message is assumed to be perfect, i.e., zero error probability and delay. The redundancy, due to symbols of reduced degree zero, is denoted \( e_0^F \) in this case.

We can divide the transmission into two phases; before feedback (phase 1) and after (phase 2). The number of symbols collected in phase 1 is denoted \( \Delta_1 \) and the total number of symbols collected in both phases is denoted \( \Delta_2 \). In phase 2 the encoder only considers refinement layer symbols, which is the equivalent of \( \beta = 0 \), thus entailing the reduced degree distribution \( \pi_0(0, \ell^X) \). Phase 1 continues as long as \( \ell^X \neq 0 \) and phase 2 continues as long as \( \ell^X \neq 0 \).

The expected redundancy, due to symbols of reduced degree zero, \( \mathbb{E}[k_0^{\ell}(\Delta_{max})] \) is found using the approach in Theorem 2. First we find the expected number of times the decoder fails in any state \( d^X \) in both phase 1 and phase 2. This provides the expected numbers, \( \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \) and \( \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \), of symbols received in any such state in the two phases. We then multiply with the probability that the next symbol is redundant, \( \pi_\beta(0, \ell^X) \), and sum over all terminal states \( d^X \) for which \( \ell^X \neq 0 \) in phase 1, since this is required for phase 1 to continue. Similarly, we multiply with \( \pi_0(0, \ell^X) \) and sum over all \( d^X \) for which \( \ell^X \neq 0 \) in phase 2. The expected redundancy is expressed in Theorem 3.

Theorem 3. (Redundancy in Two-Layer UEP LT Code with Feedback) In a two-layer UEP LT code using any degree distribution, \( \pi(i) \), and a single feedback message when the base layer has been decoded, the expected total redundancy, \( \mathbb{E}[k_0^{F}(\Delta_{max})] \), due to reduced degree zero, is:

\[
\mathbb{E}[k_0^{F}(\Delta_{max})] = \sum_{\ell^X: \ell^X \neq 0} \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \pi_\beta(0, \ell^X) + \sum_{\ell^X: \ell^X \neq 0} \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \pi_0(0, \ell^X),
\]

where \( f_{D^X}(d^X|\Delta_1) \) and \( f_{D^X}(d^X|\Delta_1, \Delta_2) \) are the terminal state distributions for phase 1 and 2, respectively.

Proof: The contribution in phase 1 follows the structure in Theorem 2, although with a different condition for receiving more symbols, since phase 1 continues when \( \ell^X \neq 0 \). The same holds for phase 2, although the amount of symbols received is \( \Delta_2 - \Delta_1 \) and the condition for receiving more symbols is \( \ell^X \neq 0 \). See proof of Theorem 2 for details.

As for Theorem 2, Monte Carlo simulations are used to evaluate \( f_{D^X}(d^X|\Delta_1) \) and \( f_{D^X}(d^X|\Delta_1, \Delta_2) \).

V. Numerical Results

We here evaluate the expressions derived in section IV. We choose \( k = 100, \alpha = 0.5 \) and the RS distribution as degree distribution, with parameters \( c = 0.1 \) and \( \delta = 1 \).

Initially, we evaluate \( \mathbb{E}[\Delta_{d^X}(\Delta_{max})] \) through Monte Carlo simulations with 1000 iterations, at different values of \( \beta \) and in the case of no feedback. We evaluate it at all possible \( D^X \) and normalize with the maximum number of received symbols, \( \Delta_{max} \), hence providing the expected fraction of symbols received in that given state. Moreover we marginalize out \( R^X \), which means we get the expected fraction as a function of \( L^X \). The results are shown in Fig. 2 for \( \beta = \{1, 4, 16, 32\} \). Note that the color code is using a logarithmic scale, in order to better visualize the results. It is seen that at \( \beta = 1 \), symbol receptions are distributed symmetrically around the line \( L^X = L^X \), which was expected. In this case, we also see that most symbols are received in states where very few input symbols have been recovered. This confirms the avalanche effect in LT decoding [10], which refers to the fact that initially only a few input symbols are recovered. Then suddenly, a single new symbol enables the recovery of all the remaining input symbols. Fig. 1 reveals that this effect is essential to the performance of standard LT codes. For higher \( \beta \) values, we see the bias towards the base layer coming into effect. However, it is also seen that the bias interferes with the avalanche effect. The avalanche fades out and new symbols are received in states where few symbols are unrecovered, leading to high probability of redundancy, cf. Fig. 1.

Next, we evaluate \( \mathbb{E}[k_0^{F}(\Delta_{max})] \) and \( \mathbb{E}[k_0^{F}(\Delta_{max})] \), again for increasing values of \( \beta \). The results are shown in
of two standard LT coded transmissions, with

Fig. 3. Normalized $E[\Delta_{\text{max}}]$ as a function of $L^L$ and $L^X$ at different $\beta$.

Fig. 2. Normalized $E[\Delta_{\text{max}}]$ as a function of $L^L$ and $L^X$ at different $\beta$.

Fig. 4. Expected redundancy due to reduced degree zero as a function of the maximum amount of collected symbols for different $\beta$ and with feedback.

Fig. 5. $E[\kappa_{\text{max}}(\Delta_{\text{max}})]$ and $E[\kappa_{\text{max}}^F(\Delta_{\text{max}})]$ as functions of $\beta$, with no limit on the amount of collected symbols, hence $\Delta_{\text{max}} = \infty$.

prioritized data has been decoded. The encoder adapts by excluding the decoded data from future encoding. Analysis reveals that such a modification dramatically decreases the redundancy in lower prioritized data.

VI. CONCLUSIONS

We have analyzed LT codes with unequal error protection, termed UEP LT codes in this letter. The analysis quantifies the expected amount of redundant symbols during a transmission. Evaluations in the case of two data segments show that this amount increases with the level of priority given to the first data segment. As a result, successful decoding of the lower prioritized data is delayed substantially. Based on this insight, a modification of the UEP LT codes has been proposed, where an intermediate feedback message informs that the higher

REFERENCES