Erschienen in: 9th Middle Eastern Simulation Multiconference : MESM 2008 ; August 26 - 28, 2008, Amman, Jordan / ed. by Marwan Al-Akaidi. Ghent : EUROSIS, 2008

Unequal error protection using LT codes and block duplication

Shakeel Ahmad*, Raouf Hamzaoui†, Marwan Al-Akaidi†
* Department of Computer and Information Science, University of Konstanz, Germany.
Email:shakeel.ahmad@uni-konstanz.de
† School of Engineering and Technology, De Montfort University, Leicester, UK.
Email:{rhamzaoui,mma}@dmu.ac.uk

***Abstract*—Luby Transform (LT) codes are powerful packet erasure codes with low encoding and decoding complexity. We provide a simple method to improve the bit error rate performance of LT codes. Moreover, we exploit our method to design a new approach for unequal error protection with LT codes. We used simulations to compare our approach with a state of the art unequal error protection technique when the information symbols are partitioned into two protection levels (most important bits and least important bits). Our approach yielded lower bit error rates for the two protection levels and lower encoding complexity at the cost of moderately higher decoding complexity.**

Keywords: Communications, channel coding, performance analysis, fountain codes, unequal error protection.

I. Introduction

Fountain codes [1], [2], [3] are probabilistic forward error correction (FEC) erasure codes with many desirable features. Whereas traditional erasure codes like Reed-Solomon codes have a fixed code rate that must be chosen before the encoding begins, Fountain codes are rateless as the encoder can generate on the fly as many encoded symbols as needed. This is an advantage when the channel conditions are unknown or time-varying because the use of a fixed channel code rate would lead to either bandwidth waste if the erasure rate is overestimated or to poor performance if it is underestimated. Compared to Reed-Solomon codes, Fountain codes have lower encoding and decoding complexity, but require a few more encoded symbols at the receiver for successful decoding.

Luby [2] proposed the first class of practical Fountain codes and called them Luby Transform (LT) codes. Shokrollahi [3] introduced another class class of Fountain codes called Raptor codes by concatenating a fixed-rate channel code with an LT code. Raptor codes have been adopted as enhanced application layer FEC by Multimedia Broadcast/Multicast System (MBMS) of the 3rd Generation Partnership Project (3GPP), IP datacast (IPDC) of Digital Video Broadcasting - Handheld (DVB-H), as well as Digital Video Broadcasting Project's (DVB) global IPTV standard.

With the growing interest in Fountain codes, the question of how to achieve unequal error protection (UEP) with these codes has been addressed [4], [5], [6]. In contrast to equal error protection (EEP) where the same level of FEC is applied to all information symbols, UEP assigns different levels of protection to different information symbols. Typically, the information symbols are protected according to their importance. UEP has been successfully used [7] for the protection of scalable image and video coders such as JPEG2000 [8] and 3D SPIHT [9]. In these works, the importance of the information symbols is expressed in terms of their contribution to overall reconstruction quality.

This paper has two main contributions. The first one is a simple method to decrease the bit error rate (BER) of LT codes [2]. The idea, which is inspired by the sliding window technique of [10], consists of virtually increasing the number of information symbols. The second contribution of the paper is a new technique for UEP with LT codes.

The paper is organized as follows. Section II contains background material about LT codes. Section III discusses a state of the art UEP technique [4] for LT codes. Section IV introduces our idea for improving the BER performance of LT codes and explains how this idea can be exploited to provide UEP. Section V presents our simulation results and shows the gains achieved by our approach.

II. Background

In this section, we explain the encoding and decoding with LT codes. More details can be found in [2].

A. Encoding

The LT encoder takes a set of k information symbols (bits or bytes, for example) and generates a potentially infinite sequence of encoded symbols of the same alphabet. Each encoded symbol is computed independently of the other encoded symbols. More precisely, given k information symbols $i_1, \ldots, i_k$ and a suitable probability distribution $\Omega(x)$ on $\{1, \ldots, k\}$, a sequence of encoded symbols $e_n,\ n \geq 1, \ldots$, is generated as follows. For each $n \geq 1$

1) Select randomly a degree $d_n \in \{1, \ldots, k\}$ according to the distribution $\Omega(x)$.
2) Select uniformly at random d_n distinct information symbols and set e_n equal to their bitwise modulo 2 sum.

The relationship between the information symbols and encoded symbols can be described by a graph (Fig. 1).

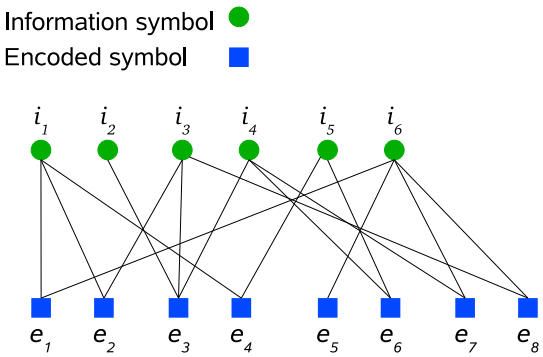

Fig. 1. Encoding graph of an LT code. Eight encoded symbols are generated from $k = 6$ information symbols. The degree of an encoded symbol is the number of information symbols that were used to generate it. For example, the degree of e_1 is equal to two.

B. Decoding

When an encoded symbol is transmitted over an erasure channel, it is either received correctly or lost. The LT decoder tries to recover the original information symbols from the received encoded symbols. We assume that for each received encoded symbol, the decoder knows the indices of the information symbols it is connected to. This is possible, for example, by using a pseudo-random generator with the same seed as the one used by the encoder.

The decoding process is as follows:

1) Find an encoded symbol e_m that is connected to only one information symbol i_j. If this is not possible, stop the decoding.
   a) Set $i_j = e_m$.
   b) Set $e_x = e_x \oplus i_j$ for all indices $x \neq m$ such that e_x is connected to i_j. Here $\oplus$ denotes the bitwise modulo 2 sum.
   c) Remove all edges connected to i_j.
2) Go to Step 1.

III. Previous work

Rahnavard, Vellambi, and Fekri [4] were the first to propose a method to provide UEP with LT codes. For simplicity, we describe their method when two levels of protection are used. Consider a source block having k information symbols. Partition these k information symbols into two sets S_1 and S_2 of size $|S_1| = \alpha k$ and $|S_2| = (1-\alpha)k$, respectively, where $0 < \alpha < 1$. The set S_1 is called the set of most important bits (MIB) while the set S_2 is called the set of least important bits (LIB). Define probabilities p_1 and p_2 ($p_1 + p_2 = 1$) to select S_1 and S_2, respectively. Given a suitable probability distribution $\Omega(x)$ on $\{1, \ldots, k\}$, a sequence of encoded symbols e_n, $n \geq 1$ is generated as follows. For each n

1) Select randomly a degree $d_n \in \{1, \ldots, k\}$ according to the distribution $\Omega(x)$.
2) Select d_n distinct information symbols successively. To select a symbol, first select one of the two sets S_1 or S_2 (S_1 with probability p_1 and S_2 with probability p_2). Then choose randomly a symbol from the selected set.
3) Set e_n equal to the bitwise modulo 2 sum of the d_n selected information symbols.

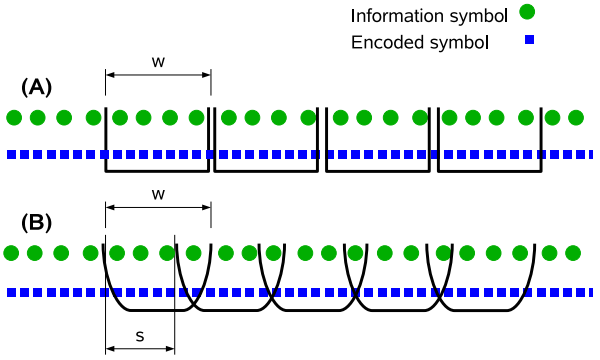

Fig. 2. Sliding window technique of [10]. (A): without window overlap. (B) with window overlap.

To ensure that the MIB symbols have lower BER than the LIB symbols, the probability of selecting an MIB symbol should be larger than that of selecting an LIB symbol [4], that is, $p_1 \frac{1}{|S_1|} > p_2 \frac{1}{|S_2|}$. To achieve this, one can set $p_1 = \frac{k_M |S_1|}{k}$ and $p_2 = \frac{k_L |S_2|}{k}$ for $0 < k_L < 1$ and $k_M = (1 - (1-\alpha)k_L)/\alpha$. Here the parameter k_M gives the relative importance of the MIB symbols.

IV. Proposed method

A. Virtual increase of source block size

We first explain our technique to improve the BER performance of an LT code. The idea is to virtually increase the size of the source block by duplicating the information symbols. This is motivated by the observation that the performance of a Fountain code improves with increasing size of the source block. Our idea is inspired by the sliding window technique proposed in [10] (see Fig. 2). This technique uses a sliding window and applies LT encoding to the information symbols within the window. The window has a size of w symbols and is shifted by s symbols until all information symbols are covered. Thus, the number of windows is $N_w = ((k-w)/s) + 1$. For example, when $s = w$, the windows do not overlap and $N_w = k/w$. When $s < w$, some information symbols are covered by more than one window. As the size of the overlap increases, the virtual size of the source block increases, resulting in higher decoding efficiency [10].

As in the method of [10], we propose to virtually increase the size of the source block. However, we do not use windows. Instead, we simply duplicate all information symbols. Simulations in Section V show that our approach gives in general better BER performance. In the following, we describe our approach in detail.

Consider a source block having k information symbols indexed from 1 to k. Let $\Omega(x)$ be the degree distribution of the LT code on $\{1, \ldots, k\}$. We expand the source block by appending the same k information symbols at

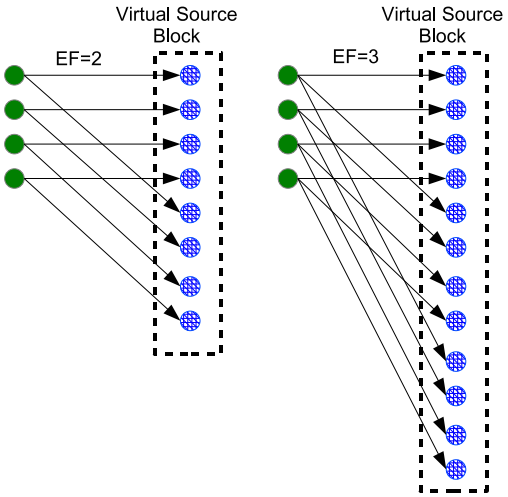

Fig. 3. Virtual increase of the source block size for $k = 4$, (left) $EF = 2$ and (right) $EF = 3$.

the end of the block. Let EF denote the number of times the source block is present in the new source block. Then the new source block has a length of $EF \times k$ and the information symbols have indices ranging from 1 to $EF \times k$ (Fig. 3). We call the parameter EF the expanding factor. Next, we extend the original degree distribution $\Omega(x)$ from $\{1, \ldots, k\}$ to $\{1, \ldots, EF \times k\}$. Using the standard LT encoder described in Section II with the new degree distribution, we generate the encoded symbols. If the index of an information symbol that is connected to an encoded symbol is between $k+1$ and $EF \times k$, we subtract $EF \times k$ from this index. In this way, the receiver can use the standard LT decoder for k information symbols.

B. Unequal error protection

The concept of virtually increasing the size of the source block by duplicating information symbols has a natural application to UEP. For simplicity and without loss of generality, we describe our UEP method for two levels of protection (MIB and LIB symbols). To realize UEP, we propose to duplicate the MIB symbols. We denote by RF the number of times a block of MIB symbols is present in the virtual source block. For example, suppose that we have six information symbols and the first two are MIB symbols. If we duplicate these two symbols as in the first step of Fig. 4, the virtual size of the source block becomes 8, corresponding to $RF = 2$. We next extend the degree distribution of the LT code from $k = 6$ to $k = 8$. To generate an encoded symbol, we find its degree d using the new degree distribution and then select d information symbols from the 8 virtual symbols. If the index of a selected information symbol is larger than 2, we map its virtual index to the actual index by subtracting 2. Finally, we note that our UEP technique can be used together with the method proposed in Section IV-A. For example, for $RF = 2$ and $EF = 2$, the source block consisting of 2 MIB symbols and 4 LIB symbols is transformed into a virtual block of size $EF(RF \times 2 + 4) = 16$ (Fig. 4).

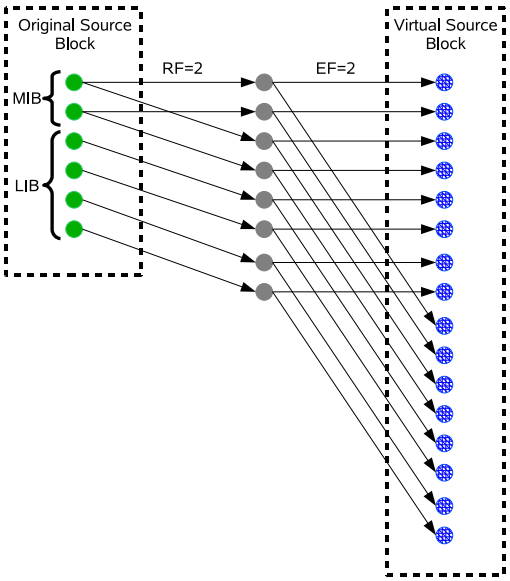

Fig. 4. UEP with $k = 6$, $EF = 2$, and $RF = 2$.

V. Experimental results

We provide two sets of experiments. The first one shows the benefits of the method proposed in Section IV-A over the sliding window approach of [10]. The second one compares our UEP technique (see Section IV-B) to that of [4] (see Section III). In all simulations, a symbol consisted of eight bits and the robust soliton distribution [2] with parameters $c = 0.1$ and $\delta = 0.5$ was used as the underlying degree distribution. The simulations were done on a PC running an AMD Dual Core 4600, with 2GB RAM.

Fig. 5 shows a comparison between our method and the sliding window approach proposed in [10]. The simulations were run for $k = 20,000$ information symbols. The overhead is calculated as $(n-k)/k$ where n is the number of encoded symbols used in the decoding and k is the number of information symbols. The BER is calculated as the average value of $(k-d)/k$, where d is the number of (correctly) decoded symbols.

The performance of the two methods improved by increasing the virtual number of information symbols. However, there was a limit beyond which no improvement was observed (50 % overlap for the sliding window approach and $EF = 8$ for our approach). The BER performance of our approach was better than that of the sliding window approach when the overhead was larger than about 0.05. When the overhead was smaller than this value, the sliding window approach gave a lower BER. However, in this range, the BER is too high for the method to be useful.

Fig. 6 compares our UEP approach to that of [4]. The performance of equal error protection is also included as a reference. As in [4], the set of information symbols was partitioned into a class of MIB symbols and a class of LIB symbols. The parameter k_M in [4] and the repeat factor RF in our approach (see Sections III and IV-B,

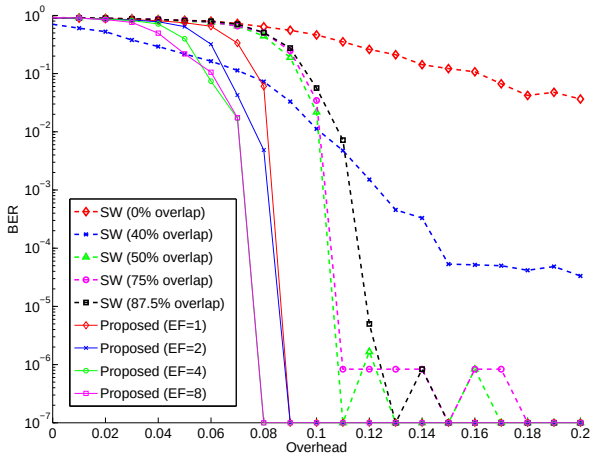

Fig. 5. Comparison between our approach and the sliding window method (SW) of [10] when the number of information symbols is $k = 20,000$. For the SW method, the window size is $w = 2,000$.

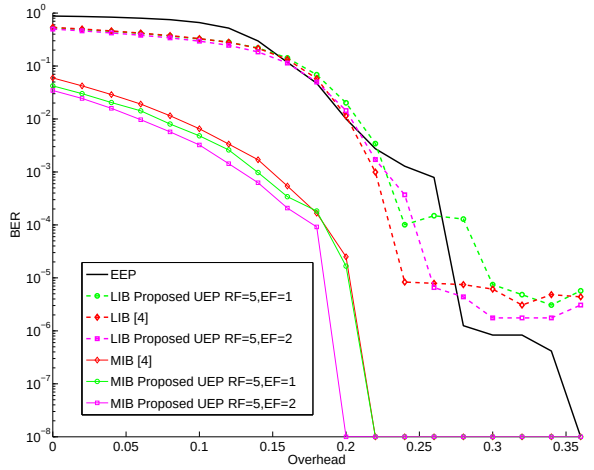

Fig. 6. BER vs. overhead for our UEP approach, that of [4], and equal error protection (EEP). There are $k = 2,000$ information symbols, 100 of which are in the MIB class.

respectively) were chosen to provide the same relative importance of the MIB with respect to the LIB. The source block consisted of $k = 2,000$ information symbols of which 100 were assigned to the MIB class.

The simulations show that with a proper choice of the parameter EF our technique can outperform that of [4] in terms of BER for both the MIB and the LIB. For example, when $EF = 2$ and $RF = 5$, our technique had a lower BER for the MIB and almost always a lower BER for the LIB. The encoding times of our technique were also lower (Fig. 7). The only penalty was in decoding times (Fig. 8).

VI. Conclusion

We presented a new method for decreasing the bit error rate of LT codes. We also exploited our approach to devise a new UEP technique for LT codes. Compared to the UEP method of [4], our technique provided lower BER for both the MIB symbols and the LIB symbols and lower encoding complexity. However, it had a slightly higher decoder complexity because of a larger average degree of the encoded symbols. It should be mentioned that the experimental work in [4] uses the optimized degree distribution of [3] rather than the robust soliton distribution used in our experiments. An interesting question is whether our UEP method also outperforms the method of [4] when the optimized degree distribution of [3] is used as the underlying degree distribution. Another topic for future research is to compare our UEP method to the UEP technique of [5]. We conclude by mentioning that by concatenating a traditional fixed-rate channel code with our UEP-LT code, we can build a UEP-Raptor code.

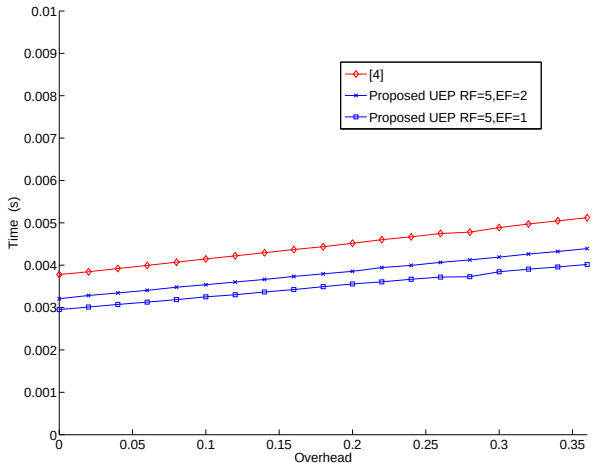

Fig. 7. Encoding times for Fig. 6.

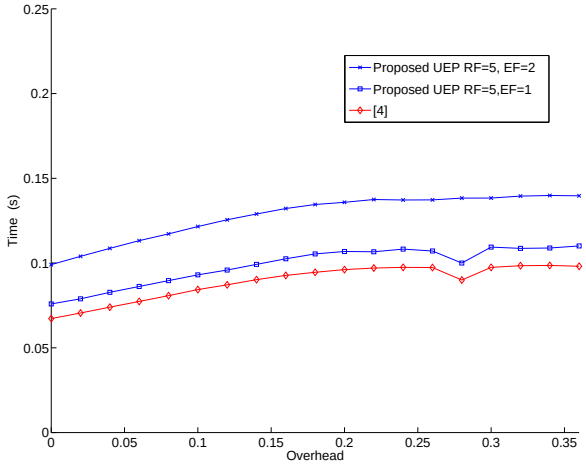

Fig. 8. Decoding times for Fig. 6.

VII. Acknowledgement

This work was supported by the DFG Research Training Group GK-1042.

REFERENCES

[1] J.W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast", IEEE Journal on Selected Areas in Communications, vol. 20, pp. 1528–1540, Oct. 2002.

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

[3] A. Shokrollahi, "Raptor Codes", IEEE Trans. Inf. Theory, Vol. 52, No. 6, June 2006, pp. 2551–2567.

[4] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property", IEEE Trans. Inf. Theory, Vol. 53, No. 53, pp. 1521–1532, April 2007.

[5] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection", Proc. 41st Asilomar Conf., Pacific Grove, pp. 1020–1024, 2007.

[6] D. Vukobratovic, V. Stanković, D. Sejdinovic, L. Stanković, and Z. Xiong, "Scalable data multicast using expanding window fountain codes", Proc. 45th Allerton Conf., Monticello, Sept. 2007.

[7] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bitstreams, IEEE Signal Proc. Mag., vol. 22, pp. 91-107, Nov. 2005.

[8] D. S. Taubman and M. Marcellin, JPEG2000: Image Compression Fundamentals, Standards, and Practice, Norwell, MA: Kluwer, 2001.

[9] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)", IEEE Trans. Circuits Syst. Video Technol., vol. 10, pp. 1365-1374, Dec. 2000.

[10] M. Bogino, P. Cataldi, M. Grangetto, E. Magli, and G. Olmo,"Sliding-window digital fountain codes for streaming multimedia contents", IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, May, 2007.

BIOGRAPHIES

Shakeel Ahmad received the B.Sc. degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2000, the M.Sc. degree in Information and Communication System from Hamburg University of Technology (TUHH), Germany, in 2005, and the Dr.-Ing degree from the University of Konstanz, Germany, in 2008. He is currently a Research Assistant at the Department of Computer and Information Science of the University of Konstanz, Germany. His research interests include multimedia communication, error-resilient video streaming and channel coding.

Raouf Hamzaoui received the Maîtrise de mathématiques from the Faculty of Sciences of Tunis, Tunis, Tunisia, in 1986, the M.Sc. degree in mathematics from the University of Montreal, Montreal, QC, Canada, in 1993, the Dr. Rer. Nat. degree from the Faculty of Applied Sciences, University of Freiburg, Freiburg, Germany, in 1997, and the Habilitation degree in computer science from the University of Konstanz, Konstanz, Germany, in 2004. He is currently a Professor of Media Technology in the School of Engineering and Technology at De Montfort University, Leicester, U.K. His research interests include image and video compression, multimedia communications, error control systems, and algorithms.

Marwan M. Al-Akaidi (M96-SM03) joined the Department of Electronic Engineering at De Montfort University in 1991. In November 2000 he became professor of Communication & Signal Processing. His main research interest is in the field of Digital Signal Processing & Digital Communications. This includes Speech Coding, Processing, Recognition, and Wireless and Mobile communication. In September 1999, Professor Marwan Al-Akaidi was appointed Chairman for the IEEE UKRI Signal Processing Society and in March 2000 as the IEEE UKRI Conferences chair. In December 2000 he was also appointed to join the board of IEEE Industrial Relations. He has won the award of the IEEE UKRI in recognition of outstanding leadership as a chapter chair for the years 2001 and 2002.