

Precoding by Priority: A UEP Scheme for RaptorQ Codes

Keshava M Elliadka
Department of Electrical Engineering
San Jose State University
San Jose, CA 95192-0084 USA
Email: keshava.elliadka@sjsu.edu

Robert Morelos-Zaragoza
Department of Electrical Engineering
San Jose State University
San Jose, CA 95192-0084 USA
Email: robert.morelos-zaragoza@sjsu.edu

Abstract—Raptor codes are the first class of fountain codes with linear time encoding and decoding. These codes are recommended in standards such as Third Generation Partnership Project (3GPP) and digital video broadcasting. RaptorQ codes are an extension to Raptor codes, having better coding efficiency and flexibility. Standard Raptor and RaptorQ codes are systematic with equal error protection of the data. However, in many applications such as MPEG transmission, there is a need for Unequal Error Protection (UEP): namely, some data symbols require higher error correction capabilities compared to others. We propose an approach that we call Priority Based Precoding Ratio (PBPR) to achieve UEP for systematic RaptorQ and Raptor codes. Our UEP assumes that all symbols in a source block belong to the same importance class. The UEP is achieved by changing the number of precode symbols depending on the priority of the information symbols in the source block. PBPR provides UEP with the same number of decoding overhead symbols for source blocks with different importance classes. We demonstrate consistent improvements in the error correction capability of higher importance class compared to the lower importance class across the entire range of channel erasure probabilities. We also show that PBPR does not result in a significant increase in decoding and encoding times compared to the standard implementation.

I. INTRODUCTION

Fountain codes [1] are a class of rateless error control codes that are suitable for applications (such as internet) in which the channels resemble binary erasure channels with a time-varying erasure probability. The encoder generates potentially unlimited number of encoding symbols (known as check or repair symbols) and the decoder can decode the source symbols from a large enough subset of these encoding symbols. Luby-Transform (LT) codes [2] are improved universal Fountain codes that have reduced encoding and decoding complexities compared to the original. Raptor codes [3] are an extension of LT codes, and are the first fountain codes with linear time encoding and decoding. They have a two stage coding scheme wherein the LT codes are concatenated with a high rate Low Density Parity-Check (LDPC) code—the *precode*—that enables the recovery of those symbols that are not recovered in the LT decoding phase. Raptor codes are used in many standard applications like Third Generation Partnership Project (3GPP) [4] Multimedia Broadcast Multicast Services (MBMS), Digital Video Broadcasting-Handheld (DVB-H) [5], etc. RaptorQ codes are the most recent class of fountain codes. They provide higher flexibility, larger source block size, and better coding efficiency compared to Raptor codes [6].

Standardized Raptor and RaptorQ codes are systematic codes and provide Equal Error Protection (EEP) to all information symbols. However, in several applications, a portion of the information symbols require higher error correction compared to others. An important example is MPEG transmission where I-frames have higher priority compared to P-frames and B-frames. This motivates the need for *Unequal Error Protection (UEP)* [7]. UEP schemes classify symbols into multiple groups, and provide different error correction to each group.

Existing UEP schemes for the Raptor family of codes can be broadly classified into two categories. The first category provides UEP by modifying the selection probability of symbols. Such schemes are applicable to customized Raptor codes [8], [9]. The second category of UEP schemes use expanding and overlapping windows of source symbols. UEP is achieved by repeating the source symbols in appropriate

windows [10].

We propose a new UEP scheme called *Priority Based Precoding Ratio (PBPR)* for standard systematic RaptorQ and Raptor codes. PBPR assumes that all symbols in a source block belong to the same importance class. The UEP is achieved by changing the number of precode symbols for the entire block based on the priority of source symbols in that block. The number of decoding overhead symbols is the same across all source blocks—even those belonging to different importance classes. A salient feature of our scheme is that it seamlessly integrates into the standard design of RaptorQ and Raptor codes.

Our work was motivated by the important application area of MPEG transmission [11]. We focus our experimental design on source block sizes that are most applicable to MPEG transmission. We show that for such block sizes, PBPR achieves consistent improvements in error correction capability for the high priority symbols compared to lower priority symbols across the entire range of erasure probabilities.

The contributions of our work are outlined below.

- 1) We construct a new UEP scheme for standard systematic RaptorQ and Raptor codes based on priority-based precoding.
- 2) Our UEP scheme demonstrates consistent and significant improvement in error correction capability for high priority symbols compared to lower priority symbols.
- 3) We empirically characterize the improvement in error correction performance as a result of increasing the number of precode symbols in RaptorQ codes.

II. BACKGROUND

RaptorQ codes are systematic codes whose failure probability curve is similar to that of random binary fountain codes. The systematic property of Raptor family of codes is achieved by creating a set of *intermediate symbols*, which are generated by “decoding” the original source symbols at encoder. These intermediate symbols are encoded to generate the original source symbols and repair symbols. The intermediate symbols are regenerated at the decoder from the set of received encoding symbols and the original source symbols are calculated from these intermediate symbols. Each symbol (both repair and source symbol) is identified by a unique 24 bit integer identifier known as *encoding symbol identifier (ESI)*, which is carried from encoder to decoder as part of the symbol overhead. The ESI is used for generating the pseudo-random distribution which is applied to calculate the repair symbol degree and to derive the set of source symbols to generate the symbol. The block diagrams showing the encoding and decoding are captured in Fig.1.

The precode of RaptorQ codes consists of two codes. An LDPC code, which is generated from a cyclically shifted circulant matrix and another dense form of the LDPC code known as Half Density Parity-Check (HDPC) code. The relationships of the intermediate symbols with the encoded symbols and constraint relations within the intermediate symbols are represented by a matrix, which is known as the *generator matrix*. Since the precoding phase is combined with the LT coding in Raptor family of codes, the generator

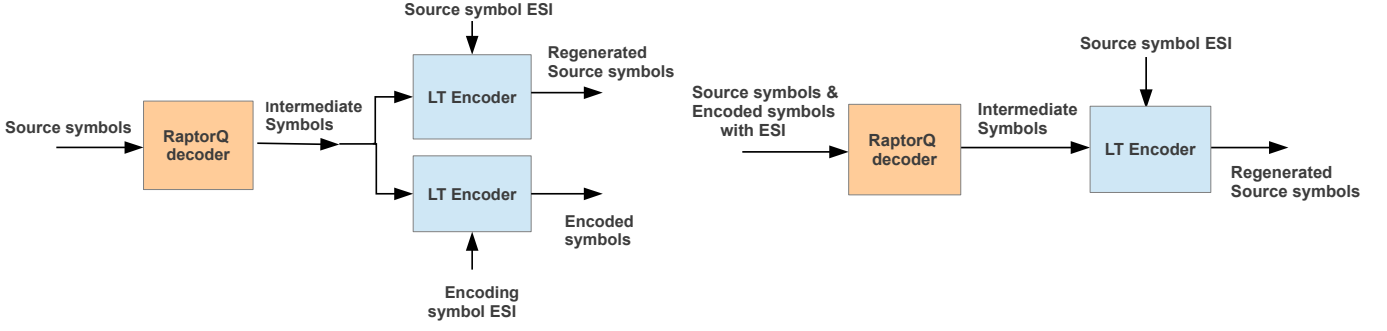


Fig. 1: Block diagrams showing RaptorQ encoding (left) and decoding (right) schemes

matrix is created from three sub-matrices. The first set of rows correspond to HDPC codes, the next set correspond to LDPC code and rest of the rows correspond to LT codes and generated from LT distribution. The schematic representation of generator matrix is shown in Fig. 2.

Raptor and RaptorQ codes use a special type of decoding method known as *inactivation decoding* [12], which is a combination of belief propagation decoding and gaussian elimination. In practice, the decoding is done by simplification of generator matrix which involves row and column reordering and row operations over \mathbb{F}_q . The overhead symbols (additional encoding symbols received by decoder) add more rows to the generator matrix, but these additional rows subsequently get dropped after choosing the set of independent rows at the end of first phase of decoding [13].

RaptorQ codes support larger block sizes, superior flexibility and better coding efficiency compared to Raptor codes due to two major enhancements in the coding scheme. First enhancement is the usage of symbols from larger finite fields. The elements of HDPC rows in RaptorQ codes are chosen from \mathbb{F}_{256} and the rest of the rows are from \mathbb{F}_2 . Using a subset of symbols from \mathbb{F}_{256} improves the failure probability curve, without causing large performance penalty [12]. Second enhancement is the change in generator matrix structure which introduces a variation to inactivation decoding known as *permanent inactivation*, wherein a subset of columns in the LT part of the generator matrix are generated with uniform random distribution.

III. RELATED WORK:UEP FOR FOUNTAIN & RAPTOR CODES

A novel scheme of UEP for Raptor family of codes was first proposed by Rahnavard et. al. [8]. It is a generic scheme that can be used for any Fountain code. This method was subsequently improved to produce different variations, such as the one proposed in [9]. The UEP is achieved by modifying the selection probability of the source symbols for check symbol generation in these schemes. The precode (LDPC encoding/decoding) and LT coding are treated as two separate

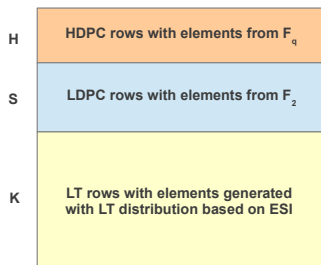


Fig. 2: RaptorQ generator matrix structure, where H is the number of HDPC symbols, S is number of LDPC symbols and K is the number of LT symbols

independent phases and UEP is applied either at LT coding (UEP-LT codes) or at precode (UEP-LDPC codes) or both. The underlying idea is to select the source symbols from higher importance classes with higher probability compared to those from lower importance classes for check symbol generation. Although such schemes are very effective, they cannot be used with standard Raptor codes which are systematic.

Another method used for achieving UEP in case of fountain codes is to divide the information symbols into multiple overlapping and expanding windows. The symbols with higher priority are made part of more number of these windows compared to symbols of lower priority, which increases the recovery probability of symbols with higher priority. This method is applied to Raptor codes by partitioning the source symbols into multiple overlapping windows to achieve better error correction [10].

UEP for systematic Raptor codes has been proposed recently with a new design for Raptor codes [14], which is different from standard design. The intermediate symbols are generated in a manner which is similar to the standard scheme with a generator matrix and these symbols are encoded using separate LDPC and LT encoding steps to generate encoding symbols. The UEP is achieved by partitioning encoding matrices into different sub-matrices and changing the properties of these sub-matrices to group the symbols into different important classes.

IV. PROPOSED UEP SCHEME: PRIORITY BASED PRECODE RATIO

We propose a new scheme of UEP for RaptorQ codes, which we call priority based precode ratio (PBPR). This scheme is designed with the assumption that all symbols in a source block (set of source symbols which participate in encoding and decoding as a unit) belong to the same importance class. The UEP is achieved by modifying the precode properties of the RaptorQ codes.

A. Key Idea

We empirically demonstrate that increasing the number of precode constraint symbols of RaptorQ codes increases the error correction capability and make use of this property to design UEP. The UEP is achieved by making source block with symbols from higher importance class to have more number of precode symbols compared to the source block with symbols from lower importance class. In our preliminary work, we modified both HDPC and LDPC symbol numbers for designing the UEP scheme. The precode constraints of Raptor codes are designed such that the symbols which are not recovered in LT decoding process are recovered by the precode with very high probability. Increasing the number of precode symbols increases the probability of recovery for those symbols which were not decoded by LT decoding. The Fig. 3 diagrammatically shows the changes introduced in the RaptorQ decoder for the design of UEP scheme.

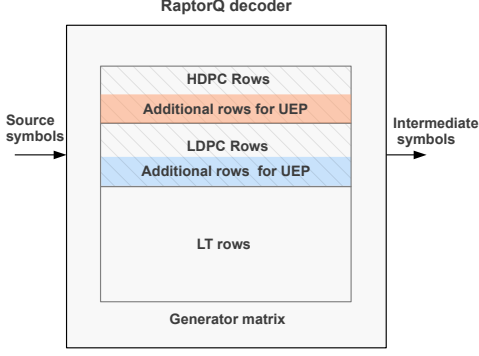


Fig. 3: RaptorQ decoder highlighting the changes done to achieve UEP by PBPR method. Number of precode rows are made proportional to the priority of symbols in source block.

B. Theoretical basis

The Raptor coding scheme relies on generation of intermediate symbols from source symbols (or from encoded symbols, in case of decoder). Once the intermediate symbols are generated, original source symbols can be regenerated based on the ESI of the source symbols. Generation of the intermediate symbols from encoded symbols is analogous to solving a set of simultaneous equations, wherein the intermediate symbols act as the independent variables which are being solved. Any method of solving linear system of equations can be applied to decode the intermediate symbols. The linear system equations can be solved only when the associated coefficient matrix is of full rank. This property makes the error correction probability of Raptor codes to be dependant on the rank profile of the generator matrix. The decoding of Raptor codes is successful if the decoder matrix is full rank and all the intermediate symbols are successfully recovered. The decoding completely fails if the matrix is full rank and none of the symbols are recovered. Therefore, all the symbols in source block which are part of decoding step have equal probability of being decoded and thus have uniform error protection.

We analyse the effect of increasing the HDPC precode symbols on the rank profile of generator matrix and show that the probability of matrix being full rank increases with increase in precode rows. In addition to improvement in the rank profile of the generator matrix, increase in the precode symbols improves the error correction capability of the RaptorQ codes in general. The precode rows of generator matrix define a set of LDPC constraint relations within the intermediate symbols. Increase in the precode rows increases the number of these constraints, which increases the probability of intermediate symbol recovery in decoding.

To analyse the effect of increasing the HDPC symbols on the rank profile of the generator matrix, let us consider a RaptorQ generator matrix A with K LT rows, H HDPC rows and N columns, as shown in Fig.4. For this analysis, we only reproduce the aspects of the derivation that we require, and refer the reader to [12] for details of the derivation. Let the probability of LT sub-matrix (K rows of the matrix) having full rank (rank of K) be p_K . This probability is same as that of standard RaptorQ codes since there is no change in this sub-matrix. Rest of the rows (HDPC precode rows) of the matrix have their elements chosen such that their properties are similar to the random binary matrix. The probability of such a matrix (matrix M_H in Fig. 4) of precode rows having full rank is given by [15], [16]

$$p_r = \prod_{i=1}^H \left(1 - \frac{1}{q^i}\right), \quad (1)$$

where q is the order of finite field, which is 256 in this case.

Let the number of LT columns (the columns corresponding to LT distribution) be W and number of permanent inactivation (PI) columns be P in the combined LT matrix M_K . The PI columns are binary and also appear as to be chosen uniformly at random. In the matrix A , K rows are considered as LT rows and H rows are the additional rows due to precode. The PI columns are the additional columns outside the LT distribution of the combined matrix M_K . Since it is an LT decoder matrix, the additional overhead on LT symbol rows due to the difference in number of rows and LT columns is described as

$$K - W = N - W - H = P - H$$

These are the additional rows which are introduced because of PI symbols. Let p_W denote the probability that the rank of the matrix M_{LT} is W . Then the rank probability of the combined matrix M_K is given by

$$p_K = p_W \prod_{i=1}^{P-H-1} (1 - 2^{-i-P}).$$

The probability p_K may be approximated by

$$p_K = p_W(1 - 2^{-H})$$

To take the additional rows (which are from \mathbb{F}_q) into account, the combined probability of matrix A having full rank can be written as

$$p_N = p_K \prod_{i=1}^H \left(1 - \frac{1}{q^i}\right) = p_W(1 - 2^{-H}) \prod_{i=1}^H \left(1 - \frac{1}{q^i}\right). \quad (2)$$

See [12] for further details of the derivation above.

As can be seen from (2), the value of p_N increases as H is increased, since the value of p_K is constant for our analysis. Since q is much larger than 2, adding terms to the second part of the equation grows much smaller than $(1 - 2^{-H})$ term. Therefore, the rank probability of matrix increases as the number of H rows is increased.

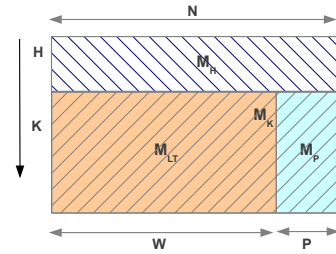


Fig. 4: Matrix A considered for rank profile analysis based on the increase in HDPC rows

C. Increase in decoding time

Increasing the precode constraints increases the number of intermediate symbols and in effect increases the number of dense rows in the RaptorQ generator matrix. These dense rows are more probable to increase the number of rows of the matrix which is used for gaussian elimination phase of decoding. Addition of rows to gaussian elimination phase increases the computation cost, since the complexity of gaussian elimination is quadratic ($O(n^2)$) compared to the linear time complexity of belief propagation. However, since the increase in precode size is relatively small compared to the number of LT symbols, the decoding time is comparable with the standard RaptorQ implementation and there is no significant increase in decoding cost.

V. EXPERIMENTAL RESULTS

The PBPR method was simulated for RaptorQ codes considering two importance classes which are commonly known as *more important bits* (MIBs) and *less important bits* (LIBs). The simulation was done for different source block sizes and different erasure probabilities. The block sizes we considered range from 53 bytes up to 213 bytes. As stated earlier, these block sizes model actual usage in MPEG transmission [11].

We followed the method proposed in [6] for simulating the encoder and decoder for RaptorQ codes, including the finite field arithmetic. The set of source symbols with block size was passed as input to the encoder. The decoder logic implemented in the encoder side generated intermediate symbols and these intermediate symbols were verified by regenerating source symbols by encoding them. The encoder generated the encoder symbols with ESI values which are greater than those which are used for source symbols. We used ESI values ranging from 0 to $2^{24} - 1$ for the encoding symbols so that sufficient randomness is introduced in the generation of LT distribution, which uses ESI value as the basis.

The erasure channel was simulated by dropping (erasing) the source symbols and encoded symbols uniformly at random with given erasure probability. Both source symbols and encoded symbols underwent same erasure mechanism to make sure that the behavior is comparable to the real-world behavior. We considered erasure values ranging from 0.1 up to 0.9 for our experiments. The decoder received the mixture of source and encoded symbols which survived the channel erasure and these symbols are used to generate the LT part of the decoder matrix based on ESI values. The precode part of the matrix was generated based on the rules proposed in [6]. Most of the simulations were done for the scenarios considering zero overhead symbols at the decoder, except for one case, in which we considered one overhead symbol. Our experiments were done with the source block sizes ranging from 55 bytes up to 213 bytes. In our preliminary investigations, we modified the precode sizes for higher importance classes such that the error correction capability is better than lower importance classes but the cost in terms of performance is not significant. Table. 1 captures the values we used for the higher importance class in comparison with lower importance class.

TABLE I: Comparison of precode symbol sizes for different source block sizes in PBPR method

K (LT Rows)	S (LDPC rows)		H (HDPC rows)	
	LIB	MIB	LIB	MIB
55	13	17	10	12
101	17	23	10	12
213	23	27	10	12

Note: Each experiment was run 10^6 times, for erasure probabilities ranging from 0.1 to 0.9 in steps of 0.1. Recall that K denotes the source block size.

A. Experiment 1: UEP for $K = 55$ bytes, zero overhead symbols

Objective. To conduct the error probability comparison experiment with source block size of 55 bytes, which was chosen from the standard [6, Table. 2].

Methodology. For the LIB source block, the precode sizes were same as suggested by standard, i.e. $H = 10$ and $L = 13$. For the MIB symbol source block we increased the HDPC precode size to $H = 12$ and LDPC precode size to $S = 17$.

Results. The results are captured in Fig. 5a showing both LIB and MIB error correction probability vs. different erasure probabilities.

Discussion. The results show that the MIB error performance is improved approximately up to 1.4 times compared to the LIB error performance.

B. Experiment 2: UEP for $K = 101$ bytes, zero overhead symbols

Objective. To conduct the error probability comparison experiment with source block size of 101 bytes, which was chosen from the standard [6, Table. 2].

Methodology. For the LIB source block, the precode sizes were same as suggested by standard, i.e. $H = 10$ and $L = 17$. For the MIB symbol source block we increased the HDPC precode size to $H = 12$ and LDPC precode size to $S = 23$.

Results. The results are captured in Fig. 5b showing both LIB and MIB error correction probability vs. different erasure probabilities.

Discussion. The results show that the MIB error performance is improved approximately up to 1.4 times compared to the LIB error performance.

C. Experiment 3: UEP for $K = 213$ bytes, zero overhead symbols

Objective. To conduct the error probability comparison experiment with source block size of 55 bytes, which was chosen from the standard [6, Table. 2].

Methodology. For the LIB source block, the precode sizes were same as suggested by standard, i.e. $H = 10$ and $L = 23$. For the MIB symbol source block we increased the HDPC precode size to $H = 12$ and LDPC precode size to $S = 27$.

Results. The results are captured in Fig. 5c showing both LIB and MIB error correction probability vs. different erasure probabilities.

Discussion. The results show that the MIB error performance is improved approximately up to 1.6 times compared to the LIB error performance.

D. Experiment 4: UEP for $K = 101$ bytes, one overhead symbol

Objective. To conduct experiments with source block size of 101 bytes and 1 overhead symbol.

Methodology. For the LIB source block, the precode sizes were same as suggested by standard, i.e. $H = 10$ and $L = 17$. For the MIB symbol source block we increased the HDPC precode size to $H = 12$ and LDPC precode size to $S = 23$. The encoding symbols used for decoding was $K + 1$ (102) in this case.

Results. The results are captured in Fig. 5d showing both LIB and MIB error correction probability vs. different erasure probabilities.

Discussion. The results show that the MIB error performance is improved approximately up to 2 times of the LIB error performance.

TABLE II: Percentage increase in encoding-decoding time with PBPR for different source block sizes

Source Block size (bytes)	Increase in Decoding Time (percentage)
55	18
101	13.35
213	7.5

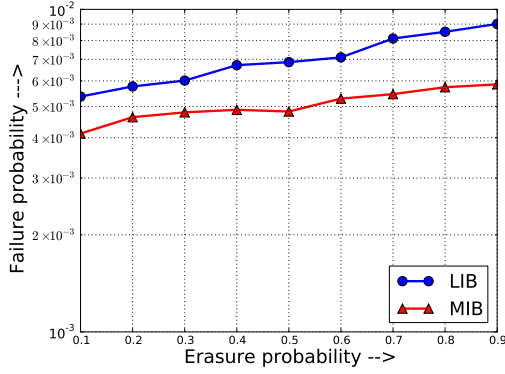
E. Experiment 5: Increase in encoding-decoding time with PBPR

Objective. To conduct experiments to measure the increase in encoding and decoding time because of increasing the precode size.

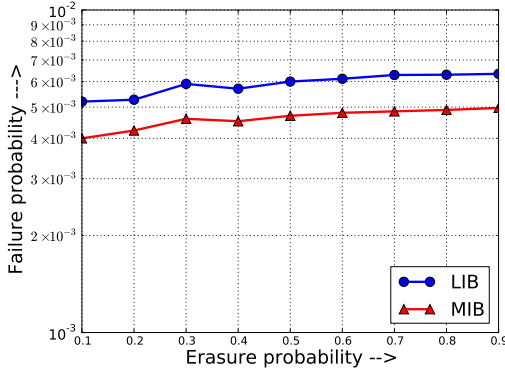
Methodology. The simulations were done for the LIB and MIB precode values specified in Table I. The time taken for simulation was measured by averaging over 5000 runs. The percentage increase in the MIB encoding/decoding time for different source block sizes are captured.

Results. The results in the form of percentage increase in time are captured in Table II.

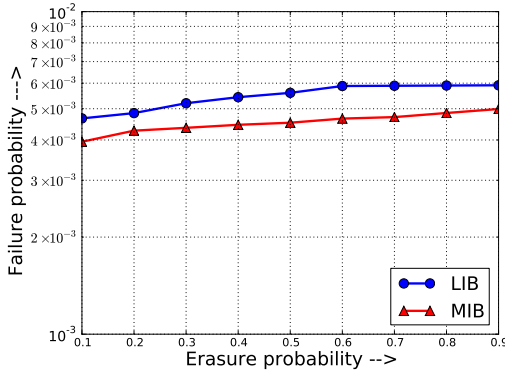
Discussion. The table shows that there is some increase in the performance, which is up to 18% increase in the time taken by MIB symbol encoding and decoding. This performance penalty is proportional to the ratio of increase in precode symbols to the LT symbols. This penalty seems to be decreasing as the ratio is decreased.



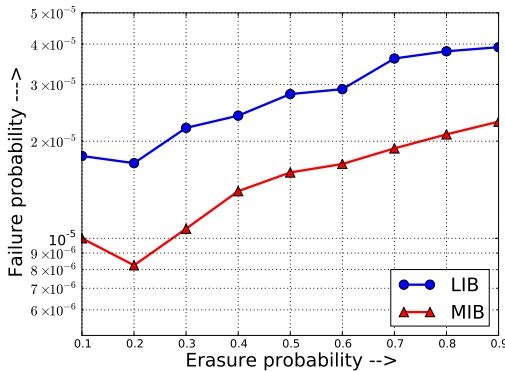
(a) Source block size of 55 bytes.



(b) Source block size of 101 bytes.



(c) Source block size of 213 bytes.



(d) Source block size of 101 bytes with 1 overhead symbol.

Fig. 5: Comparison of MIB and LIB performance for different source block sizes. The failure probability of MIB and LIB symbols is shown for different erasure probabilities from 0.1 to 0.9 in steps of 0.1.

VI. CONCLUSION

We propose a new UEP scheme called PBPR for RaptorQ and Raptor codes. PBPR is designed with the assumption that all symbols in a source block belong to same importance class. UEP is achieved by changing the number of precode symbols based on the priority of symbols in source block. We empirically show that increasing the number of precode symbols increases the error correction capability of RaptorQ codes. Our experimental results confirm a consistent and significant increase in error correction capability of higher importance symbols compared to lower importance ones. We also show that the increase in encoding and decoding time using PBPR is insignificant compared to standard implementations.

In the longer version of the paper, we will report on a significantly expanded experimental design, and also include a complete theoretical treatment for the improvement of error correction capability witnessed by using PBPR.

ACKNOWLEDGMENT

Authors wish to thank Nazanin Rahnavard for graciously providing helpful pointers to literature.

REFERENCES

- [1] D. J. C. MacKay, "Fountain Codes," *IEEE Communications*, vol. 152, pp. 1062–1068, 2005.
- [2] M. Luby, "LT Codes," in *Foundations of Computer Science, FOCS 2002*. IEEE, 2002, pp. 271–282.
- [3] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [4] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs," 3rd Generation Partnership Project (3GPP), TS 26.346, Dec. 2004. [Online]. Available: <http://www.3gpp.org/ftp/Specs/html-info/26346.htm>
- [5] DVB, "Digital Video Broadcasting (DVB); IP Datacast over DVB- H: Content Delivery Protocols," European Broadcasting Union, TS 102.472, Jun. 2009. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102400_102499/102472/01.03.01_60/ts_102472v010301p.pdf
- [6] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "RaptorQ Forward Error Correction Scheme for Object Delivery," RFC 6330 (Proposed Standard), Internet Engineering Task Force, August 2011. [Online]. Available: <http://www.ietf.org/rfc/rfc6330.txt>
- [7] B. Masnick and J. Wolf, "On linear unequal error protection codes," *Information Theory, IEEE Transactions on*, vol. 13, no. 4, pp. 600–607, 1967.
- [8] 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.
- [9] L. Yuan and J. An, "Design of uep-raptor codes over bec," *European Transactions on Telecommunications*, vol. 21, no. 1, pp. 30–34, 2010.
- [10] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. J. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.
- [11] 3GPP, "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs," 3rd Generation Partnership Project (3GPP), TS 26.234, Dec. 2013. [Online]. Available: <http://www.3gpp.org/DynaReport/26234.htm>
- [12] M. A. Shokrollahi and M. Luby, "Raptor codes," *Foundations and Trends in Communications and Information Theory*, vol. 6, no. 3–4, pp. 213–322, 2009.
- [13] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," RFC 5053 (Proposed Standard), Internet Engineering Task Force, October 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc5053.txt>
- [14] K. Noh, S. Kim, and J. Heo, "Design of systematic unequal error protection raptor codes over binary erasure channels," *Wireless Personal Communications*, pp. 1–10, 2013.
- [15] C. Cooper, "On the distribution of rank of a random matrix over a finite field," *Random Struct. Algorithms*, vol. 17, no. 3–4, pp. 197–212, 2000.
- [16] J. Blmer, R. M. Karp, and E. Welzl, "The rank of sparse random matrices over finite fields," *Random Struct. Algorithms*, vol. 10, no. 4, pp. 407–419, 1997.