An Efficient Algorithm for Encoding and Decoding of Raptor Codes over the Binary Erasure Channel*

ZHANG Ya-Hang[1,2+], CHENG Bo-Wen[1,2++], ZOU Guang-Nan [1], WEN Wei-Ping[2], QING Si-Han[2]
1(Space Star Technology Co.,Ltd, Beijing 100081, China)
2(Department of Information Security, SSM, Peking University, Beijing 100084, China)
+ Corresponding author: Phn: +13488694401, E-mail: zhangyahang@gmail.com
++ Corresponding author: Phn: +13488694376, E-mail: shuanshui_66@163.com

ABSTRACT

As the most advanced rateless fountain codes, Systematic Raptor codes has been adopted by the 3GPP standard as a forward error correction scheme in Multimedia Broadcast/Multicast Services (MBMS). It has been shown to be an efficient channel coding technique which guarantees high symbol diversity in overlay networks.

The 3GPP standard outlined a time-efficient maximum-likelihood (ML) decoding scheme that can be implemented using Gaussian elimination. But when the number of encoding symbols grows large, Gaussian elimination need to deal with a large matrix with O (K^3) binary arithmetic operations, so the larger K becomes, the worse ML decoding scheme performs.

This paper presents a better time-efficient encoding and decoding scheme while maintaining the same symbol recoverable performance, this encoding and decoding scheme is named Rapid Raptor Code. It will be shown that the proposed Rapid Raptor code Scheme significantly improves traditional Raptor codes' efficiency while maintaining the same performance.

1. INTRODUCTION

RAPROT Codes [1] has recently been the most advanced error-correcting codes over the binary ensure channel (BEC). When given a set of message symbols, Raptor codes can generate a finite length of encoding symbols and are able to recover the set of message symbols perfectly from any subset of encoding symbols whose cardinality is only slightly greater than that of the source symbols. Furthermore, Raptor codes can be encoded and decoded in linear-time which makes it very appealing for various applications. For these reasons, Raptor codes has recently been adopted by the 3GPP standard as the forward error correction scheme in Multimedia Broadcast/Multicast Services (MBMS) [2][4].

In this paper, we present a better time-efficient encoding and decoding schemes than traditional Raptor codes while maintaining the same symbol recoverable performance. In encoding scheme, the improved encoding algorithm combines the pre-encoding step and LT Encoding step to one step. After that one generator matrix called Matrix Z is generated. Z is a b×k matrix over GF(2), in which each row corresponds to a repair symbol and each column to a source symbol and the ith repair symbol is equal to the sum of those input symbols whose column contains a non-zero entry in row i. Then Z matrix is used to generate b repair symbols from k source symbols directly.

In decoding schema, assumed that there is k' source symbols and b_1' repair symbols received. At the beginning, each of the b_1' repair symbols decrease its degree by exclusive-ORing the data portion of the received source symbols which were used to calculate this repair symbol in encoding step. After this step, assume that there are b_2' ($b_2' \leq b_1'$) left. Then the b_2' repair symbols only have the relationship with k-k' source symbols which were lost during transmission. A $b_2' \times$ (k-k') matrix Z' over GF(2) is generated to represent this relationship of repair symbols and (k- k') source symbols. Finally, the (k-k') lost source symbols can be recovered by triangulating Z' using Gaussian elimination.

Without the need for calculating the intermediate symbols before generating repair symbols, the improved encoding schema is simpler and more efficient and the improved decoding schema compute a much smaller matrix which size is

International Conference on Space Information Technology 2009, edited by Xingrui Ma, Baohua Yang, Ming Li,
Proc. of SPIE Vol. 7651, 765134 · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.855395

associated with the product of the erasure probability of the binary erasure channel (BEC) and the number of source symbols k. That makes both encoding and decoding algorithms better time-efficient and less memory consuming as compared to the original decoding scheme defined in 3GPP Standard.

2. TRADITIONAL RAPTOR CODES

2.1 Traditional Raptor Codes Encoding Schema

Traditional Raptor codes encoding schema [3] are cascaded codes with a pre-code and the LT code. There are two steps in Systematic Raptor codes encoding phase. For a k-symbol message m, the first step is to generate a number L (L> k), of intermediate symbols from this k source symbols. And the relationship between L and K meets L = K+S+H. Here, s and h are the amount of redundancy added by the pre-code step. After this step, k source symbols are extended to L pre-code symbols. The 'generator matrix' M for pre-code that generates L intermediate symbols from L pre-code symbols is an L*L matrix over GF (2), where it is a combination of LDPC, Half and LT encoding generator matrixes. And we have (1) in the first step. In the second step, L pre-codes symbols will produce a N repair symbols by applying a 'encoder matrix' R where each row corresponds to one of the repair symbols and each column to one of the pre-codes symbols and where the ith output symbol data is exclusive-ORed from those intermediate symbols whose column contains a non-zero entry in row i. And N is the computing result for system redundancy. Hence we have (2) in the second step. The encoding schema defined as follows:

$$M = G1_{L\times L} \times C \tag{1}$$

$$R = G2_{N\times L} \times M \tag{2}$$

2.2 Traditional Raptor Codes Decoding Schema

The decoding scheme suggested in the 3GPP MBMS standard is a version of the computationally expensive “standard” Gaussian elimination decoding and consists of four phases. The first phase is to convert the original matrix A into a matrix consisting of left upper and lower and right submatrix. In this matrix, the left upper submatrix is a diagonal matrix and the left lower submatrix is an all-zero matrix. The submatrix V which is formed by the intersection of all but the first i columns and the last u columns and all but the first i rows of A for nonnegative integers i and u. The matrix V will change during the course of the first phase at the end of which will disappear if this phase is success. [3]Gaussian elimination is performed in the second phase on the left lower submatrix to either determine that its rank is less than u (decoding failure) or to convert it into a matrix where the first u rows is the identity matrix (success of the second phase). After the second phase, the only portion of A that needs to be zeroed out to finish converting A into the L by L identity matrix is the left upper submatrix. In the third phase, for each of the first i rows of A, and for each group of 8 columns in the upper submatrix of this row, if the set of 8 column entries in the upper are not all zero, then the row of the pre-computation matrix U' that matches the pattern in the 8 columns is exclusive-ORed into the row, thus zeroing out those 8 columns in the row at the cost of exclusive-ORing one row of U' into the row. In the fourth phase, A is the L by L identity matrix and a complete decoding schedule has been successfully formed. We can calculate the intermediate symbols from the received symbols. Once intermediate symbols are generated, the missing source symbols can be recovered from intermediate symbols using exclusive-OR operation [4].

3. RAPID RAPTOR CODES

3.1 Rapid Raptor Codes Encoding Schema

As described above, a K-symbol message is first used to generate an L-symbol intermediate message, and the matrix $G1_{L\times L}$ whose column contains a non-zero entry in row i represents that the corresponding source symbol is exclusive-ROed to the *ith* intermediate symbol. And Figure 1 shows the composition of $G1_{L\times L}$, I_S is the *S x S* identity matrix, I_H is the H×H identity matrix, and 0_S×H be the S×H zero matrix.

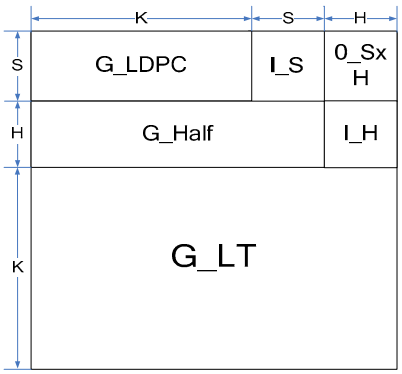

Fig. 1 Composition of $G1_{L\times L}$

In the second step, the matrix $G2_{N\times L}$ is generated from *LTEnc* function, and the ith row whose column is 1, represents the corresponding intermediate symbol is exclusive-ORed to the ith repair symbol. Since we can calculate and in advance, and from (1) and (2). We can simply deduce the equation (3), we have matrix $A_{N\times L}$ and let $A_{N\times L} = G2_{N\times L} \times G1_{L\times L}$. Make use of matrix $Z_{N\times K}$ to represent the last K column of Matrix $A_{N\times L}$, then the equation (4) can be formed from equation (3)

$$R = G2_{N\times L} \times G1_{L\times L} \times C \tag{3}$$

$$R = Z_{N\times K} \times C \tag{4}$$

Hence we have (4) that represents the relationship between K source symbols and N repair symbols. And it means we no longer need to calculate intermediate symbols in encoding step. The relationship between Z and A shows in figure 2.

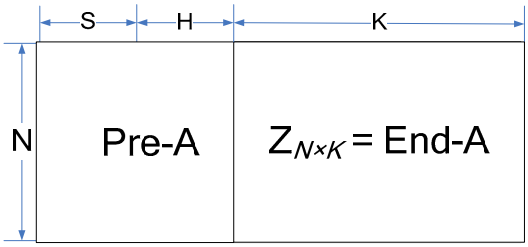

Fig. 2 Relationship between Z and A

In Rapid Raptor codes Encoding Schema, we combined the pre-encoding step and LT Encoding step to one step. In combined step one generator matrix called Matrix Z is generated. Matrix Z is a $b \times K$ matrix over GF(2), in which each row corresponds to a repair symbol and each column to a source symbol and the ith repair symbol is equal to the sum of those input symbols whose column contains a non-zero entry in row i. Then Z matrix is used to generate b repair symbols from K source symbols directly. Apparently, the degree of the ith repair symbol is the number of the columns contains a non-zero entry in row i; the degree of each source symbol is 1.

3.2 Rapid Raptor Codes Decoding Schema

Since we do not involve intermediate symbols in encoding phase, we also need not to recover intermediate symbols in our decoding schema. Suppose the receiver received K' source symbols and N' repair symbols (K' <= K, N' <= N). The number of lost source symbols over BEC channel is $k = K-K'$, the number of lost repair symbols is $N-N'$. In our decoding schema, repair symbols are collected to source symbols, since its data is exclusive-ORed from numbers of source symbols and matrix Z represents their relationship. We call the number of source symbols which are involved to generate the *i*th repair symbol is the degree of this repair symbol is d_i. And the degree set of all repair symbols is $d = (d_1, d2 \ldots d_{N'})$,

while the degree of the entire source symbols is 1. In our decoding schema, we use all the received source symbols to exclusive-ORed all the repair symbols to reduce the degree [6] of repair symbols. If the *i*th symbol d_i contains the *j*th symbol in K', the *i*th repair symbol should be exclusive-ORed with the *j*th source symbol. The following algorithm named "Degree Reduce Algorithm" described as followed:

```
1:repeat
2:      new receive node s
3:      if s is repair symbol then
4:              if isUseful(s) then
5:                      put into result-list R
6:              else
7:                      abandon s
8:      else
9:              put into check-list C
10:until no node received
11:for all r in R do
12:     for all c in C do
13:             if r contains c then
14:                     remove c from r
15:             end if
16:     end for
17:end for
```

In which the *isUseful(*s*)* function is used to judge if the received repair symbol is exclusive-ORed by the any lost source symbols by querying the generate matrix $Z_{N\times K}$.

After "reduce degree" step, assume there are still n ($k<=n<=N'$) useful repair symbols left, the vector of which is $R'=(R'_1, R'_2 \ldots R'_n)$. Every repair symbol in R' is viewed as a combination of lost source symbols. And their degree set is a subset of *d*. We convert their degree collection to an $n\times k$ matrix Z', the *i*th row of $Z'_{n\times k}$ can be viewed as the data of *i*th repair symbol is exclusive-ORed from those lost source symbols whose column contains 1. All the lost source symbols satisfy (5) with R'.

$$R' = Z'_{n\times k} \times C' \tag{5}$$

In the following step, we use "Gaussian elimination" to solve small matrix Z'. Apparently, if Z' is full rank, we can recover the lost source symbols from those degree-reduced repair symbols.

4. COMPLEXITY ANALYSIS

4.1 Theoretical Analysis

A. Encoding Process

In out approach, all the work should do in the encoding step is to operate the equation (4). For a fixed length message we can save a fixed Matrix Z without doing operations on matrix, so the time needed in the encoding process is equal to the overhead ofβmessage. Given a message with k symbols, the time encoding complexity of Rapid Raptor Code is $\beta*O(K)$ compared to $(1+\beta)*O(K)$ of traditional Raptor codes.

B. Decoding Process

The biggest improvement is made in the decoding process. As described in section 3.1, repair code is generated according to equation (4), so apparently we can recover all of the source symbols from the repair symbol using ML decoding algorithm [6] based on Gaussian elimination to solve small matrix Z, this scheme need to deal with the matrix Z, and This task can be done in polynomial time using Gaussian elimination. However, Gaussian elimination is not fast enough, especially when the matrix Z is too large (the time cost of Gaussian elimination is $3K^3+2K^2+K$, where *K* is the size of the matrix Z).

As describe in section 3.2, we first let all the received repair symbols degreed with the source symbols received already. After this step, the repair symbols left are only related with the lost source symbols. Then the small matrix $Z'_{n\times k}$ is formed with $k = K\text{-}K'$ columns, which much smaller than the original generate matrix $Z_{N\times K}$, than we can recover all of the lost source symbols from the left useful repair symbol using ML decoding algorithm, then the time cost of this decoding process is $3k^3+2k^2+k$. The time cost of "Degree Reduce Algorithm" is k, so the total time cost of this decoding process T = $3k^3+2k^2+k+k = 3k^3+2k^2+2k$, where $k(= K\text{-}K')$ is the size of the matrix Z', and this is a big progress as compared to decoding scheme of traditional Raptor.

4.2 Practical Result

In this section, we present the encoding and decoding performances obtained using traditional Raptor codes defined in [2][3] and Rapid Raptor codes scheme. We have realized the two types of codes using C on Linux (kernel version 2.6). All simulations in this paper were tested on Intel(R) Pentium(R) CPU @2.33GHz. Specially, in our implements, the size of each symbol is 16384 byte, the CPU and the Operating System is 32-bit.

A. Encoding Process

The results of encoding process are based on 10000 runs with encoding overhead equal to 10%, 5%, 2% and 1%. In Figure 3, the length of the message is K = 1031 (which means the size of data is 1031×16384 bytes).

Figure 3 shows time cost of the encoding process of traditional Raptor codes and Rapid Raptor codes. As shown in Figure 3, in encoding step, the Rapid Raptor codes time cost is almost proportional to the encoding over head while traditional Raptor codes is not very sensitive to the change of encoding over head. That is because traditional Raptor codes need to calculate a fix number ($L = K+S+H$) of immediate codes at pre-code step while Rapid Raptor codes need not to, and this step cost too much time. So the Encoding complexity of traditional is much bigger than that of Rapid Raptor codes, especially when the encoding overhead is small, and source symbol number is large.

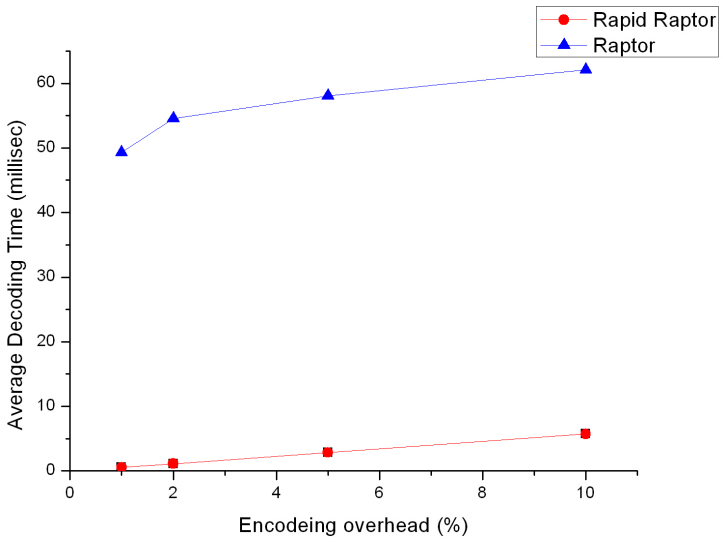

Fig.3. Encoding times in milliseconds (K=1031).

B. Decoding Process

The results of decoding process are based on 10000 runs with encoding overhead equal to 20%, the code lost rate equal to 10%, 5%, 2%, 1%, 0.5%. In Figure 5 the length of the message is K = 1031 (which means the size of data is 1031× 16384 bytes).

Figure 4 shows the Time Cost used to deal Matrix by decoder in milliseconds; Figure 5 shows time cost of the decoding process of traditional Raptor codes and Rapid Raptor codes. As shown in these Figure 4 and Figure 5, in decoding step,

the Rapid Raptor codes time cost is almost proportional to the cube of encoding over head while traditional Raptor codes is not very sensitive to the change of encoding over head. As described in section 3.2, the reason is because in traditional code, the decoder needs to deal with a bigger L×L matrix through a time-efficient ML decoding scheme defined in 3GPP, and need to calculate the immediate codes which have relationship with the lost symbols, while in our Rapid Raptor codes, the decoder just need to deal with a much smaller matrix $Z'_{n\times k}$ to calculate the lost source symbols directly without recover the immediate codes, where the $k=\beta*K$. So, as the k becomes smaller, the less time will be taken by the ML decoding algorithm.

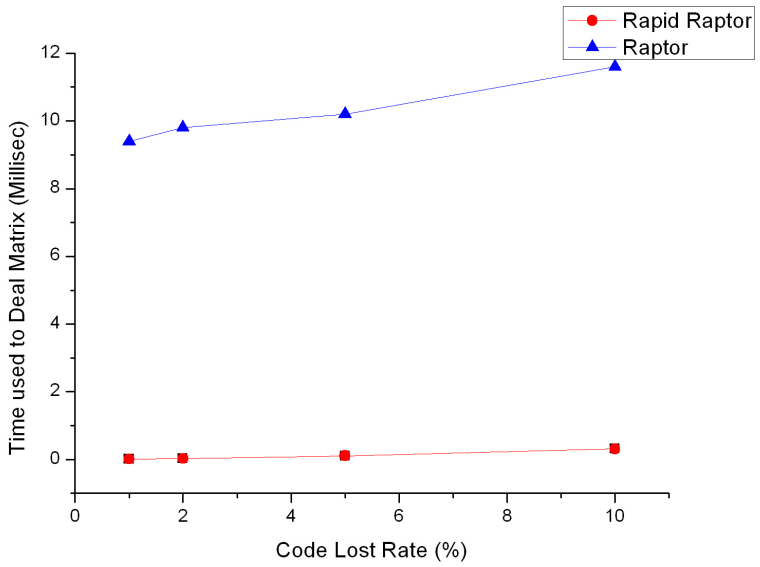

Fig. 4 Times Cost used to deal Matrix by decoder in milliseconds (K=1031).

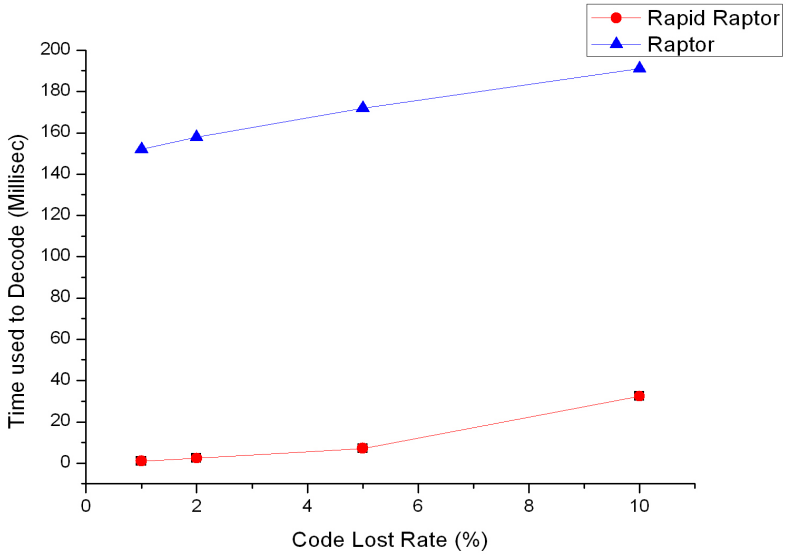

Fig. 5 Decoding times in milliseconds (K=1031).

5. CONCLUSION

In this paper, an improved Raptor codes scheme--- Rapid Raptor codes was shown. Without the need of calculating the intermediate symbols before generating repair symbols, the improved encoding algorithm in Rapid Raptor codes is much simpler and more efficient, while the improved decoding algorithm only to compute a much smaller matrix which size is associated with the product of the erasure probability of the binary erasure channel (BEC) and the number of source symbols k. That makes both encoding and decoding algorithms better time-efficient and less memory consuming as compared to the original Raptor Code scheme defined in RFC5053 [3] and 3GPP [2].

REFERENCES

1. A. Shokrollahi, "Raptor Codes," IEEE Trans. Inform. Theory, vol. 52, no.6, pp. 2551–2567. June 2006.
2. 3GPP TS 25.346 v.7.0.0, 2006, Technical Specification Group Radio Access Network; Introduction of the Multimedia Broadcast/Multicast
3. Services (MBMS) in the Radio Access Network, Mar. 2006.
4. M. Luby, "Raptor Forward Error Correction Scheme for Object Delivery," www.ietf.org/rfc/rfc5053.txt, IETF, 2004
5. Saejoon Kim, Seunghyuk Lee, Sae-Young Chung, "An Efficient Algorithm for ML Decoding of Raptor Codes over the Binary Erasure Channel," IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 8, AUGUST 2008.
6. Saejoon Kim, Karam Ko, Sae-Young Chung, "Incremental Gaussian Elimination Decoding of Raptor Codes over BEC," IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 4, APRIL 2008
7. M. Luby, "LT Codes," in Proc. 43rd Annual IEEE Symp. Foundations of Computer Science, Vancouver, Canada, 2002.