Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA-13)

# Analysis and Implementation of Raptor Codes for Video Multicast


Wei Geng
College of Computer & Information, Hohai University, Nanjing, 210098, China
wei_geng@163.com

Li Congjuan
School of Information Science & Engineering, Southeast University, Nanjing, 210096, China
blue_gw@163.com

Zhang Yudong
MRI Lab, Columbia University, New York, NY, 10032, USA
zhangyudongnuaa@gmail.com



***Abstract*—Packet loss is unavoidable in IP multicast using the UDP protocol that will serious influence video quality. In this paper we use the Raptor codes for video multicast to improve the video quality. Firstly, the principle of Raptor codes is analyzed in detail. Then the Raptor codes is implemented and transplanted in the video transmission system. Simulation results show that the video transmission system with Raptor FEC works which can overcome the packet loss effectively and improve the playback quality significantly.**



***Keywords-Raptor codes; multicast; VLC; VTN; video coding***


## I. INTRODUCTION

IP multicasting technology is widely used in video services, such as video conferencing and IPTV. Packet loss is unavoidable in IP multicast using the UDP protocol, which will seriously affect the playback quality. Common methods used to overcome packet loss include ARQ, Data Carousel and Erasure Codes. Due to poor scalability and large delay, ARQ is not suitable for video multicast, neither is data carousel which cycles through and sends video packets. Traditional Erasure Codes including simple XOR codes, RS codes and Tornado codes are just with limited repair capacity because of poor scalability, complexity and other drawbacks. Digital fountain codes are flexible, scalable and rateless codes. Raptor codes [1][2] are the second practical realizations and one kind of Raptor codes has been incorporated into the recent standards, namely 3GPP MBMS (Multimedia Broadcast/Multicast Services) [3] and DVB-H [4]. Therefore, this paper applies it to the video multicast.

Raptor codes are quite useful for forward error correction (FEC) since they have the advantages of linear time encoding and decoding. Furthermore, they own the ideal code performance under any channel loss condition. In recent years, implementation and performance evaluation of Raptor codes for multimedia applications have been studied [5]. As far as we know, analysis and implementation of Raptor codes for video multicast on their software implementation have not yet been investigated.

This paper looks at the implementation and transplant of Raptor codes for video multicast system by using software method. The organization of the paper is as follows. Section 2 analyzes the principle of Raptor codes. Section 3 describes the implementation and transplant of Raptor codes on some video system in detail. The simulation results are presented in Section 4. Section 5 concludes the paper.

## II. RAPTOR CODES

Generally speaking, the coding process includes two steps. The first step is pre-coding which generates the *L* intermediate symbols by using *K* source symbols ( $L > K$ ). The second step is LT coding to generate the final repair symbols by using the intermediate symbols.

### A. Pre-coding

The pre-coding can be shown in Fig.1.

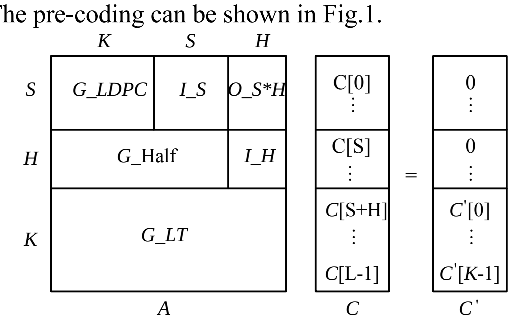


Fig.1 Pre-coding.

In Fig.1, A is a square matrix whose elements are 1 and 0. The C'[0], C'[1],..., C'[K-1] are K source symbols and C[0], C[1],..., C[L-1] are L intermediate symbols. The source symbol triples are generated so that for any K matrix, A has full rank and is therefore invertible. Thus, the calculation of L intermediate symbols can be realized by applying a Raptor decoding process to the K source symbols.

$$C = A^{-1} * C' \qquad (1)$$

The source symbol triple is an important concept in the Raptor. The K source symbol triples are associated with the K source symbols and are then used to determine the L intermediate symbols. Each of the K source symbols is associated with a triple (d[i], a[i], b[i]) for 0 <= i < K. The source symbol triples are determined using the Triple generator defined as follow.

For each i, 0 <= i < K

(d[i], a[i], b[i]) = Trip [K, i]

L is defined as follows.

$$L = K + S + H \qquad (2)$$

where S is the smallest prime integer such that S >= ceil(0.01*K) + X. X is the smallest positive integer such that X*(X-1) >= 2*K. H is the smallest integer such that choose (H, ceil(H/2)) >= K + S.

Published by Atlantis Press, Paris, France.

It should be noted that the L intermediate symbols are the uniquely defined symbol values that satisfy the following two conditions given the K source symbols:

(a) The K source symbols satisfy the K constraints

C'[i] = LTEnc[K, (C[0],..., C[L-1]), (d[i], a[i], b[i])], for all i, 0 <= i < K.

where LTEnc is the LT encoder.

(b) L intermediate symbols includes S LDPC symbols and H Half symbols. The S LDPC symbols and its previous K symbols satisfy the following relationships.

```
For i = 0,..., K-1 do
    a = 1 + (floor(i/S) % (S-1))
    b = i % S
    C[K + b] = C[K + b] ^ C[i]
    b = (b + a) % S
    C[K + b] = C[K + b] ^ C[i]
    b = (b + a) % S
    C[K + b] = C[K + b] ^ C[i].
```

### B. LT Encoding

In LT encoding, the repair symbol with ESI X is generated by applying the generator LTEnc[K, (C[0], C[1],..., C[L-1]), (d, a, b)] to the L intermediate symbols C[0], C[1],..., C[L-1] using the triple (d, a, b)=Trip[K,X].

### C. Decoding of Raptor codes

The decoding is the inverse process of coding. Firstly, the intermediate symbols are obtained after performing LT coding. Then the source block including K source symbols is recovered after decoding the intermediate symbols.

## III. IMPLEMENTATION AND TRANSPLANT OF RAPTOR CODES

### A. Encoding

The encoding processing is performed on the video server, i.e., Video Transmission Network (VTN) developed by Huawei Co. Ltd [6]. The flow chart of coding can be shown in Fig. 2. The coding includes four steps: coding parameter configuration and computation, coding, packet encapsulation, and packet sending. In order to speed up the process, the three steps, i.e., coding, packet encapsulation and sending are implemented in different thread.

*1) Parameter configuration and computation*

In the configuration file, the initial values of the parameters are set as follows: $K$=128, $R$=32, $G$=1, where $K$ denotes the number of source symbols. $R$ denotes the number of repair symbols for a source block. The bigger R means the bigger redundancy of coding which means more strong recovery capability. G denotes the number of symbols in a package. According to the coding process, the coding parameters can be calculated as follows: $S$=19, $H$=10, $L$=147, and A is a 147×147 square matrix in the Galois field with 2 elements, i.e., GF(2). Moreover, the *G_LDPC*, *G_Half* and *G_LT* are obtained [2].

*2) Packet encapsulation*

In this paper, the RTP/UDP is accepted. As for the *.TS file, the largest payload for each RTP package is set to be 1316 bytes which includes seven TS packages. As for the *.264 file, the largest payload for each RTP package is set to be 1316 bytes, as well as 20 bytes of heads, the longest of a RTP is 1336 bytes. A RTP package is set to be a source symbol. A source block includes K RTP packages and is set to be a coding unit.

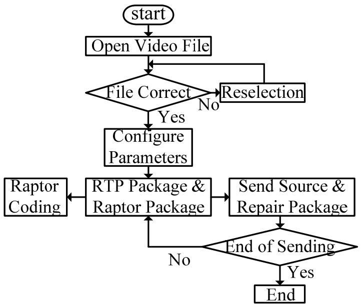


Fig.2. Flow chart of Raptor coding.

*3) Coding realization*

The Raptor coding module can be shown in Fig.3. The process includes four steps. Firstly, the number of the source package ,i.e. N, is checked. If $N \geq K$ , the coding block including K source packages from the buffer is obtained. The coding block is the source symbol array $C'$ . Then the pre-coding is performed and consequently the intermediate symbol array is obtained according to Equ.(1). After that the intermediate symbols are input to the LTEnc and the repair symbols are generated by LT coding.

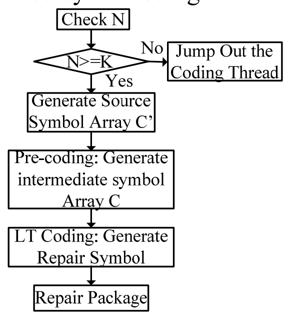


Fig.3. Flow chart of Raptor coding module.

*4) Packet sending*

The package send is performed after the repair packages are obtained. Given the memory consumption and delay, the source package is required to synchronize with the repair package.

### B. Decoding

The decoding is the inverse process of coding. The decoding processing is performed on the VideoLAN Client (VLC) [7] which is an open source and cross platform video player. The decoding process of Raptor sub-thread can be shown in Fig.4.

The decoding process can be summarized as follows. After receiving the Raptor package, the decoder firstly check whether there are package losses and record the information of losses. Then the lost packages are recovered by Raptor decoding. At last the recovered packages are inserted to the corresponding RTP queue and the source packages are decoded.

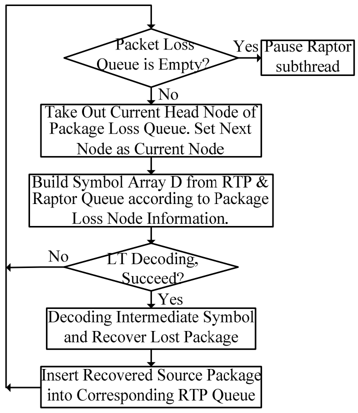


Fig.4. Decoding process of Raptor sub-thread.

## IV. SIMULATION RESULTS

To evaluate the performance of Raptor codes for video multicast, the detailed encoding specifications used in the experiments are listed below:

(1) Sending end: Windows 2000 Server
(2) Multicast address: 224.10.12.12
(3) Receiving end: Windows XP
(4) Video player: VLC-0.9.8
(5) Software used for lost package: wipfw-0.2.8
(6) Source Sequence: Amazon.yuv (720×480 format)
(7) Bitrate: 1Mbps

We use the remainder packet loss ratio (RPLR) as the objective measurement to evaluate the performance of Raptor codes for video multicast. The RPLR can be defined as follow:

$$RPLR = \frac{[\overline{A}]-[\overline{R}]}{2000} \times 100\% \quad (3)$$

where A and R mean the number of actual and recovered lost package, respectively. $\overline{X}$ means the average of X. In this simulation, we count the number 40 times. [X] is an integer which is not larger than X. The number, i.e. 2000, means that a test unit includes 2000 packages.

Table 1 shows the recovered results of lost packages after Raptor decoding. From this table, we can see that the RPLR is very small. The results indicate that Raptor can reduce the PLR greatly in practice. Moreover, the recovery capability of Raptor codes strengthens with the enlargement of K. For example, the RPLR is only 0.05% when K=512 and actual PLR=13%.

We also use the classical 5-level method [8] to subjectively evaluate the performance of Raptor codes. The 5-level scale is described as follows.

(1) A: Imperceptible
(2) B: Perceptible but not annoying
(3) C: Slightly annoying
(4) D: Annoying
(5) E: Very annoying

Table 2 shows the results of the evaluation. We can see that the decoded video quality is slightly annoying when PLR=1% without Raptor decoding (K/R=0/0 means decoding without Raptor decoding). When the PLR=5%, the video quality is very annoying and consequently unacceptable. The quality varies with the K/R. When K/R=512/128 and PLR=14%, the video quality is only slightly annoying using Raptor decoding (K/R≠0/0). The simulation results indicate the Raptor codes improve performance significantly.

## V. CONCLUSION

Online video services such as video conferencing and IPTV are using IP multicasting technology. Packet loss is unavoidable in IP multicast using the UDP protocol, which will seriously affect the playback quality. This paper adopts Raptor codes to overcome the packet loss for video multicast. Simulation results show that the Raptor FEC can overcome the packet loss effectively and improve the video quality greatly.

## ACKNOWLEDGMENT

This research work is financially supported by the Fundamental Research Funds for the Central Universities of China (2011B09514), the Natural Science Foundation of Chongqing City (No. CSTC, 2008BB2313) and the Huawei Innovation Fund (BON2008031). The authors thank the anonymous reviewers for their constructive comments.

## REFERENCES

[1] A. Shokrollahi, “Raptor codes,” IEEE Trans. Inf. Theory, vol. 52, June 2006, pp. 2551–2567.

[2] M.Luby, A.Shokrollahi, M.Watson, etal. IETF RFC5053: Raptor Forward Error Correction Scheme for Object Delivery [DB/OL]. http://www.ietf.org/rfc/rfc5053.txt?number=5053, Oct 2007.

[3] 3GPP TS 26.346, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, 3GPP Technical Specification, Rev. V7.4.1, June 2007.

[4] Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols, ETSI Technical Specification, Rev. V1.2.1, 2006.

[5] P. Cataldi, M. P. Shatarski, M. Grangetto, and E. Magli, “Implementation and performance evaluation of LT and raptor codes for multimedia applications,” in Proceedings, IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP’06, 2006.

[6] http://www.huaweidevice.com/worldwide/

[7] The homepage of VLC [EB/OL]: http://www.videolan.org/

[8] Subjective Video Quality Assessment Methods for Multimedia Applications. ITU-T Recommendation P.910, Recommendations of the ITU, Telecommunication Standardization Sector.

Table 1. Recovered results of Raptor decoding given different coding parameters

| K/R \ PLR | 32/8 | | | 64/16 | | | 128/32 | | | 256/64 | | | 512/128 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\overline{A}$ | $\overline{R}$ | RPLR (%) | $\overline{A}$ | $\overline{R}$ | RPLR (%) | $\overline{A}$ | $\overline{R}$ | RPLR (%) | $\overline{A}$ | $\overline{R}$ | RPLR (%) | $\overline{A}$ | $\overline{R}$ | RPLR (%) |
| **1%** | 21 | 21 | 0 | 21 | 21 | 0 | 20 | 20 | 0 | 20 | 20 | 0 | 21 | 21 | 0 |
| **2%** | 40 | 40 | 0 | 41 | 41 | 0 | 41 | 41 | 0 | 40 | 40 | 0 | 40 | 40 | 0 |
| **3%** | 63 | 62 | 0.05 | 62 | 62 | 0 | 62 | 62 | 0 | 65 | 65 | 0 | 65 | 65 | 0 |
| **4%** | 85 | 82 | 0.15 | 85 | 85 | 0 | 83 | 83 | 0 | 81 | 81 | 0 | 81 | 81 | 0 |
| **5%** | 104 | 102 | 0.1 | 108 | 107 | 0.05 | 105 | 105 | 0 | 103 | 103 | 0 | 106 | 106 | 0 |
| **6%** | 127 | 121 | 0.3 | 127 | 126 | 0.05 | 128 | 128 | 0 | 127 | 127 | 0 | 127 | 127 | 0 |
| **7%** | 152 | 135 | 0.85 | 147 | 146 | 0.05 | 149 | 149 | 0 | 152 | 152 | 0 | 153 | 153 | 0 |
| **8%** | 170 | 147 | 1.15 | 173 | 171 | 0.1 | 174 | 174 | 0 | 176 | 176 | 0 | 174 | 174 | 0 |
| **9%** | 203 | 169 | 1.7 | 200 | 192 | 0.4 | 200 | 200 | 0 | 197 | 197 | 0 | 197 | 197 | 0 |
| **10%** | 217 | 178 | 1.95 | 226 | 217 | 0.45 | 217 | 217 | 0 | 222 | 222 | 0 | 222 | 222 | 0 |
| **11%** | ---- | ---- | ---- | 249 | 232 | 0.85 | 253 | 252 | 0.05 | 247 | 247 | 0 | 248 | 248 | 0 |
| **12%** | ---- | ---- | ---- | 280 | 198 | 4.1 | 274 | 272 | 0.1 | 278 | 276 | 0.1 | 279 | 278 | 0.05 |
| **13%** | ---- | ---- | ---- | ---- | ---- | ---- | 295 | 289 | 0.3 | 297 | 286 | 0.55 | 299 | 298 | 0.05 |
| **14%** | ---- | ---- | ---- | ---- | ---- | ---- | 322 | 281 | 2.05 | 330 | 306 | 1.2 | 322 | 306 | 0.8 |
| **15%** | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | 357 | 326 | 1.55 |

K/R: the parameters of Raptor codes.

---- : means that we can't count the number of actual and recovered lost package given some K/R when PLR is too large.

Table 2. Subjectively evaluation on Raptor codes.

| K/R \ PLR | 0/0 | 4/1 | 8/2 | 16/4 | 32/8 | 64/16 | 128/32 | 256/64 | 512/128 |
|---|---|---|---|---|---|---|---|---|---|
| **1%** | C | B | B | A | A | A | A | A | A |
| **2%** | D | C | C | B | A | A | A | A | A |
| **3%** | D | D | C | B | B | A | A | A | A |
| **4%** | D | D | D | B | B | A | A | A | A |
| **5%** | E | E | D | B | B | A | A | A | A |
| **6%** | E | E | E | C | C | B | A | A | A |
| **7%** | E | E | E | D | C | B | A | A | A |
| **8%** | E | E | E | D | D | B | A | A | A |
| **9%** | E | E | E | D | D | C | A | A | A |
| **10%** | E | E | E | E | D | C | A | A | A |
| **11%** | E | E | E | E | E | C | B | A | A |
| **12%** | E | E | E | E | E | D | C | B | B |
| **13%** | E | E | E | E | E | E | C | C | B |
| **14%** | E | E | E | E | E | E | D | D | C |
| **15%** | E | E | E | E | E | E | E | E | D |