Accelerated Expanding-Window FEC Using RS-Code for Real-Time Video Streaming

Conference Paper · December 2019

DOI: 10.1109/HotICN48464.2019.9063209

Sheng Cheng
Peking University

Xinggong Zhang
Peking University

Zongming Guo
Yantai Nanshan University

Accelerated Expanding-window FEC Using RS-Code for Real-time Video Streaming

Sheng Cheng, Lemei Huang, Xinggong Zhang, Zongming Guo
Wangxuan Institute of Computer Technology
Peking University
Email: {chaser_wind, milliele, zhangxg, guozongming}@pku.edu.cn

I. Introduction

Real-time video streaming is taking off in recent years. Video is predicted to account for 82% of Internet traffic by 2022, a growing share of which will take the form of live streaming [1]. Forward Error Correction (FEC) [2] is commonly applied to protect transmitted data in real-time video streaming, by sending redundant parity packets to recover the lost packets on the spot.

Expanding-window FEC [3], [4] is proposed for real-time video streaming to provide Unequal Error Protection to packets in different frames, i.e. I frames and P frames. In expanding window approach, encoding window is expanding each time the parity packets are generated (Fig.1). For example, the f-th group of parity packets are generated after all source packets from frame 1 to f are collected. For simplicity, we assume frames are with equal size, but it works for unequal-sized frames as well.

On the one hand, by enlarging the encoding window, Expanding-window FEC increases the error-correcting capability of RS code, compared with Packet- [5] or Frame- [6], [7] level FEC.

On the other hand, it also provides unequal protection for different frames, compared with Sliding-window FEC [8], as the front frames (e.g. P-frame) of a GOP are more important than others.

However, expanding-window FEC poses two challenges to practical implementation, both of which may significantly degrade the perceived visual quality of the received video:

1) **High computation cost:** Expanding-window FEC increases the number of source packets to be encoded, and thus the computation cost of both encoding and decoding.
2) **Long Decoding Delay:** To get the lost packets recovered, the decoder needs to wait for the whole block of RS code to arrive, which results in a decoding delay.

In this paper, we practically implement expanding-window FEC using RS-code (EW-RS) for real-time video streaming. We accelerate both the encoding and decoding process by reducing non-zero elements in coding matrix while maintaining the high error-correcting capability, and reduce the decoding delay by taking out solvable part of equation in advance. Evaluations shows that about 40-50% computation cost and decoding time can be saved in our implementation.

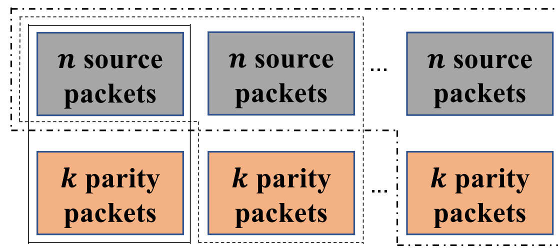


Fig. 1. Expanding Window FEC

II. Design of Low-cost EWRS

A. Implementation

EW-RS can be generally described as follows. Let l be the packet size (padding to l if not) . Assume a coding block includes f frames and i-th frame consists of n_i source packets ($1 \leq i \leq f$). k_i parity packets $P_{k_i \times l}$ are generated by (1) from $(\sum_{j=1}^{i} n_j)$ source packets $S_{(\sum_{j=1}^{i} n_j)\times l}$ and sent after each frame.

$$P_{k_i \times l} = E^{(i)}_{k_i \times (\sum_{j=1}^{i} n_j)} \times S_{(\sum_{j=1}^{i} n_j)\times l} \tag{1}$$

As for decoding, the i-th group of source packets $S_{(\sum_{j=1}^{i} n_j)\times l}$ can be decoded when expanding window decoding condition [4] holds by jointly solving the equations (**??**) for all frames prior to i-th frame. According to received source packets and parity packets, useless rows can be eliminated and thus a non-singular matrix equation is ready to be solved.

Considering the overall process, the **coding matrix** $E_{(\sum_{j=1}^{f} k_j)\times(\sum_{j=1}^{f} n_j)}$ used is in the form of a lower block-triangular matrix.

$$E_{(\sum_{i=1}^{f} k_i)\times(\sum_{i=1}^{f} n_i)} = \begin{bmatrix} E^{(1,1)}_{k_1\times n_1} & \mathbf{0} & ... & ... & \mathbf{0} \\ E^{(2,1)}_{k_2\times n_1} & E^{(2,2)}_{k_2\times n_2} & \mathbf{0} & ... & \mathbf{0} \\ ... & ... & ... & ... & ... \\ E^{(f,1)}_{k_f\times n_1} & E^{(f,2)}_{k_f\times n_2} & ... & ... & E^{(f,f)}_{k_f\times n_f} \end{bmatrix}$$

where $E^{(i,j)}_{k_i\times n_j}$ refers to the j-th sub-matrix of i-th encoding matrix $E^{(i)}_{k_i\times(\sum_{v=1}^{j} n_v)}$, i.e.

$$E^{(i)}_{k_i\times(\sum_{v=1}^{j} n_v)} = [E^{(i,1)}_{k_i\times n_1} \; E^{(i,2)}_{k_i\times n_2} \; ... \; E^{(i,i)}_{k_i\times n_i}]$$

Since EW-RS significantly enlarges the coding matrix, computation cost is greatly increased especially in decoding when solving a matrix equation.

We save computation cost by reducing non-zero elements in coding matrixes. In our implementation, we let each $E^{(i,j)}_{k_i \times n_j}$ be a band matrix, which may increase the probability of decoding failure. We explore the trade-off between them in Sec. III-A and show that the computation cost can be reduced by carefully deciding the band width of matrix.

B. Real-time Decoding

In our implementation, the unit of decoding is coding window i.e. frame $1 \sim i$, also namely i-th coding window. The i-th coding window is decodable if and only if no more than $\sum_{v=j}^{i} k_v$ packets are lost from j-th frame to i-th frame for all integer j $s.t.$ $1 \le j \le i$. Note that both source packet loss and parity packet loss are considered. We're able to calculate the maximal decodable coding window by checking decodability condition anytime, so decoding will be tried each time a packet is received. If decodable coding window lengthens, we can get the sub-matrix of coding matrix corresponding to the coding window, solve the sub-equation and recover lost packets.

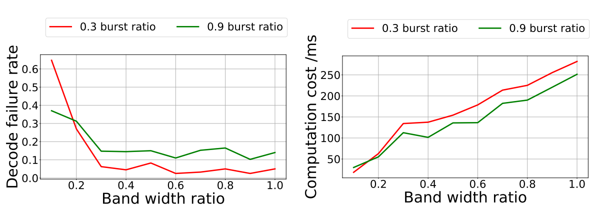
For example in equation.2, assume each frame contains only one source packet, $source_1$ was recoverable by solving sub-matrix $parity_1 = a \times source_1$ as long as $parity_1$ is obtained which makes 1-st coding window decodable.

$$\begin{bmatrix} parity_1 \\ parity_2 \end{bmatrix} = \begin{bmatrix} a & 0 \\ c & d \end{bmatrix} \times \begin{bmatrix} source_1 \\ source_2 \end{bmatrix} \tag{2}$$

III. Evaluation

A. Decodability vs Decoding Time

We tune the band width of each band matrix $E^{(i,j)}_{k_i \times n_j}$ (denoted by the proportion w_r to matrix row size k_i), and test the performance under different network situations. Let each frame contain 20 source packets and 20 parity packets ($n = n_i = 20, k = k_i = 20\ \forall i \in [1, f]$), and a GOP includes 10 frames. Average packet loss rate is 0.3, and average length of burst legnth is $0.3k$ and $0.9k$ Results are shown in Fig. 2. A satisfying value of w_r is about 0.5, accelerating the decoding process by about 40% while decoding failure rate increases within 10%.



(a)Low burst (b)High burst

Fig. 2. Performance with diverse band width

B. Error-correct & Decoding Time

We compare the performance of our EW-RS implementation with two baseline FEC method: *Frame-level FEC* and *GOP-level FEC*, both of which are using RS-code. The evaluation are launched in different network state with different packet loss probability, and high($\approx k$) or low($<< k$) burst length. The configuration of n, k and GOP length remains the same as in Sec.III-A. Results are in Fig.3. In terms of error-correction capability, EW-RS outperforms Frame-level FEC with larger coding window, and in terms of decoding time, EW-RS are less than GOP-level FEC, which validates that our implementation successfully balances the trade-off: it is more effective than Frame-level FEC and more practical than GOP-level FEC.

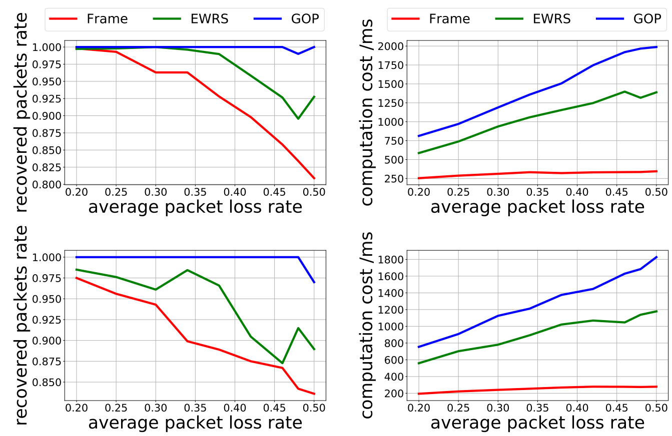


Fig. 3. Overall Performance

IV. Conclusion

In this paper, we propose expanding-window FEC using RS-code (EW-RS) for real-time video streaming. We accelerate decoding significantly by reducing non-zero elements in coding matrix while maintaining high error-correcting capability, and make it possible to decode packets online by taking out the solvable part of matrix equation.

References

[1] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper", 2019, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[2] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[3] J. Xiao, T. Tillo *et al.*, "Real-time video streaming using randomized expanding reed–solomon code," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1825–1836, 2013.

[4] Y. Geng *et al.*, "Unequal error protection for real-time video streaming using expanding window reed-solomon code," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 3763–3767.

[5] Razavi *et al.*, "Adaptive packet-level interleaved fec for wireless priority-encoded video streaming," *Advances in Multimedia*, p. 3, 2009.

[6] Hsiao, Ping-Wen *et al.*, "An analysis modeling of frame-level forward error correction for h. 264/avc over burst-loss channels," in *2014 4th ICWC, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*. IEEE, 2014, pp. 1–5.

[7] Y. Sohn, J. Hwang *et al.*, "Adaptive packet-level fec algorithm for improving the video quality over ieee 802.11 networks," *International Journal of Software Engineering and Its Applications*, vol. 6, no. 3, pp. 27–34, 2012.

[8] Bogino, Mattia *et al.*, "Sliding-window digital fountain codes for streaming of multimedia contents," in *2007 IEEE International Symposium on Circuits and Systems*. IEEE, 2007, pp. 3467–3470.