Adaptive Hybrid Forward Error Correction Coding Scheme for Video Transmission

XIONG Yuhui[1], LIU Zhilong[2], XU Lingmin[2], HUA Xinhai[2], WANG Zhaoyang[1], BI Ting[1], JIANG Tao[1]
(1. Huazhong University of Science and Technology, Wuhan 430074, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202402011
https://kns.cnki.net/kcms/detail/34.1294.TN.20240428.1442.002.html, published online April 28, 2024
Manuscript received: 2023-07-22

Abstract: This paper proposes an adaptive hybrid forward error correction (AH-FEC) coding scheme for coping with dynamic packet loss events in video and audio transmission. Specifically, the proposed scheme consists of a hybrid Reed-Solomon and low-density parity-check (RS-LDPC) coding system, combined with a Kalman filter-based adaptive algorithm. The hybrid RS-LDPC coding accommodates a wide range of code length requirements, employing RS coding for short codes and LDPC coding for medium-long codes. We delimit the short and medium-length codes by coding performance so that both codes remain in the optimal region. Additionally, a Kalman filter-based adaptive algorithm has been developed to handle dynamic alterations in a packet loss rate. The Kalman filter estimates packet loss rate utilizing observation data and system models, and then we establish the redundancy decision module through receiver feedback. As a result, the lost packets can be perfectly recovered by the receiver based on the redundant packets. Experimental results show that the proposed method enhances the decoding performance significantly under the same redundancy and channel packet loss.

Keywords: video transmission; packet loss; Reed–Solomon code; Kalman filter

Citation (Format 1): XIONG Y H, LIU Z L, XU L M, et al. Adaptive hybrid forward error correction coding scheme for video transmission [J]. ZTE Communications, 2024, 22(2): 85–93. DOI: 10.12142/ZTECOM.202402011
Citation (Format 2): Y. H. Xiong, Z. L. Liu, L. M. Xu, et al., "Adaptive hybrid forward error correction coding scheme for video transmission," *ZTE Communications*, vol. 22, no. 2, pp. 85–93, Jun. 2024. doi: 10.12142/ZTECOM.202402011.

1 Introduction

Mobile video services have proliferated with the growth of wireless communications and the Internet, and video traffic had been expected to account for 82% of total network traffic by 2022. However, networks with limited capacity are struggling to support the growing number of mobile video users. The complex uncertainty of wireless channels limits transmission rates, while wired transmissions suffer from packet loss due to buffer congestion at routing nodes. For the former, there are well-known channel coding methods[1–3] that promise transmission rates close to the Shannon bound. For the latter, existing solutions mainly involve retransmission techniques or forward error correction coding. The retransmission[4] does not require any redundant packets, but the extra round-trip time (RTT) increases the end-to-end delay of the entire video and therefore does not guarantee real-time video transmission. Forward error correction (FEC) coding[5], by way of contrast, is of interest due to its ability to recover lost source packets without adding any RTT.

The well-known WebRTC uses an exclusive OR (XOR)-based[6] FEC coding that generates new redundant packets by XORing the original packets. These redundant packets are sent to the receiver together with the original packets and the receiver recovers the lost packets according to the corresponding mapping relationships. There should be neither too many nor too few redundant packets, as this would result in waste or inadequate protection. Therefore, the FEC should adjust the number and size of redundant packets to the network environment to balance reliability and latency. To select the appropriate level of redundancy to cope with dynamic network environments, the WebRTC uses the current redundancy state to query the FEC redundancy for the next packet. The XOR encoding and single-step adaptive algorithms described above together form the FEC scheme for WebRTC.

Other FEC schemes also focus on improving packet loss recovery through advanced encoding methods and adaptive algorithms. Examples of such methods include fountain codes[7], Raptor codes[8–9], and Reed-Solomon (RS) codes[10–11]. Among them, RS codes are widely used by the telecommunication industry due to their superior protection capabilities. However, the drawback of RS codes is that decoding requires multiple matrix inversions, which results in extremely high computational complexity. In situations where the matrix dimension is too high, such as in high-definition video transmission with large packets, the decoding time may be too long to meet the needs of real-time transmission.

As for other adaptive algorithms, ATIYA et al. introduced a non-linear prediction method for automatic feature selection[12], and EMARA et al. combined an ingenious coding scheme with a network adaptive algorithm for parameter updating[13]. However, these approaches relied solely on historical network patterns to predict future patterns, overlooking the complex relationships that may exist between past and future patterns. To better exploit the correlation between current and previous network states in a weak network environment, CHENG et al. proposed a DeepRS[14] adaptive redundancy control algorithm in 2020. The algorithm uses a long short-term memory (LSTM) network[15–16] to predict the probability of packet loss and dynamically adjusts the redundancy rate of the RS encoder. However, integrating the LSTM network with the underlying user datagram protocol (UDP) protocol requires significant engineering effort.

In response to the shortcomings of existing coding and adaptive algorithms, we propose a new coding scheme covering low-density parity-check (LDPC) and RS to ensure smooth transmission of arbitrary definition video and design a multi-step Kalman filter-based adaptive algorithm for practical deployment. The contributions in this paper are summarized as follows.

• We develop a hybrid FEC highly efficient encoding method to cope with continuous burst packet loss and different application scenarios. We design an encoding scheme of the LDPC code and the RS code in their respective optimal code length ranges. We optimize a progressive edge growth algorithm to get the LDPC coding matrix of the application layer. The coding scheme of the design system could cover various source code lengths and reduce the computational complexity of codes.

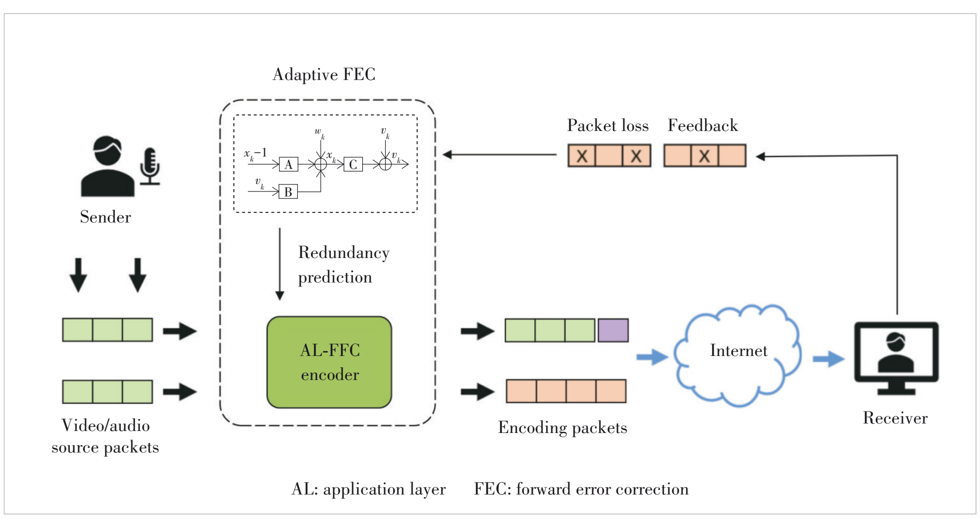
• We propose a Kalman filter-based multi-step adaptive method for video transmission. The system makes multi-step predictions based on packet loss feedback and then predicts the coding code rate based on the decoding terminal redundancy. It turns out that our method always maintains a high data recovery ratio with the interval change of the packet loss rate.

The rest of this paper is organized as follows. Section 2 describes the system architecture. Next, Section 3 proposes a frame-level partitioning method. Then, we present the hybrid coding method in Section 4. The code rate adaptation algorithm is evaluated in Section 5. Sections 6 and 7 give the evaluation and conclusions.

2 System Architecture

The prototype system verification framework for combating weak network conditions in video and audio transmission is shown in Fig. 1. The system uses the application-layer end-to-end FEC technology at the video frame level, and selects the optimal bit rate based on the network status information fed back by the terminal. Metrics such as historical packet loss rates and throughput are used to predict future network states and adaptive packet loss compensation is performed to optimize overall performance.

The sender serves as the video input source to the system, providing raw video streams that are processed into multimedia stream files with frame structures through generic protocol encoding. Before transmission, the sender side establishes a channel for transmission by selecting the optimal bitrate based on the network status information feedback from the terminal. To improve the reliability of end-to-end mobile video transmission, the sender-side encoder performs frame-level FEC (we give a frame-level partitioning method in this reign as shown in Section 3) encoding on the multimedia stream at the application layer and packages and sends the data according to the real-time transport protocol (RTP)/UDP principles. Specifically, on a small timescale, the application-layer FEC encoder processes each video frame in a fast serial manner based on the coding rate, that is, it divides each frame into source data packets, performs FEC encoding on these packets to generate repair data packets that facilitate the recovery of the original video data stream by the receiver. The mobile terminal serves as the video output and processes the received data packets by unpacking them. After obtaining the raw data packets with the RTP/UDP headers removed, the terminal-side decoder performs decoding and error correction accord-



▲Figure 1. Framework of FEC

ing to the agreed FEC encoding and decoding principles and finally obtains the raw video stream through the decoder.

Upon receiving the restored video stream, the mobile terminal sends real-time feedback on network status information (such as packet loss rate and throughput) through a control signaling port to the adaptive module located at the sender. The adaptive module monitoring and prediction unit in the module uses the feedback information to determine the encoding rate of the FEC encoder on the sender in the next slot. It is worth noting that during this process, the adaptive module needs to prevent excessive coding redundancy that may cause transmission congestion, while ensuring that the FEC encoder generates enough repair data packets to support data packet recovery, without compromising user experience in weak network environments. Additionally, the application-layer FEC encoder on the sender should always maintain the same data packet generation, validation, and recovery as the application-layer FEC decoder on the mobile terminal.

3 Frame-Level Partitioning

We divide the video into blocks according to the frames. Each block contains one or more frames, and we ensure the FEC encoding and decoding are synchronized with the video timestamp as much as possible. The sender obtains the video frame information by calling the FFmpeg tools and recombines the frames into blocks. The FEC encoding and decoding process at the application layer is based on the entire block. Successful decoding can obtain the entire video data of the block. In this paper, we set a limit of K_{frame} for the number of frames in a block, as excessively long blocks cause additional video delays. After the video is divided into blocks, we divide the blocks into coding data packets.

For short codes, the upper limit of the block size is 20× 1 400 bytes, where 1 400 setups are based on the maximum transmission unit (MTU) and 20 based on the decoding performance of RS coding (we use RS coding because of its superior performance in this region as is shown in Section 4.2). Generally, the data sizes of B-frames and P-frames are small[17]. We take several consecutive B-frames or P-frames as a block, if the data size of the block does not exceed 28 000 bytes and the number of frames in the block does not exceed K_{frame}. The block data carries the encoding method and block size (in bytes). Optionally, we can place it in the block header as basic information.

The optimal interval selection $[N_{\min}, N_{\max}]$ for medium-long codes is more flexible. Due to the encoding characteristics of the LDPC code, longer code length results in better decoding performance. In addition, different code lengths and encoding code rates correspond to different LDPC generator matrices, where the storage complexity of LDPC generator matrices is $O(n^2)$. For instantaneous decoding refresh (IDR) frames that contain a large amount of information, they can be used directly as a block. Accordingly, we combine several P-frames with a large number of data into a single block, when the number of frames is not larger than K_{frame} and the block data size is not larger than $N_{\max}$×1 400 bytes. Within the allowable range of decoding delay, a longer code length means an enhanced ability to cope with continuous packet loss.

4 Hybrid Coding Method

In this section, we first present an improved LDPC that balances decoding latency and error correction performance in application layer packet loss scenarios. The optimal operating regions of RS and LDPC codes are then designed based on the performance analysis of the improved LDPC and RS codes. The two coding methods are combined to cover the requirements of various code lengths.

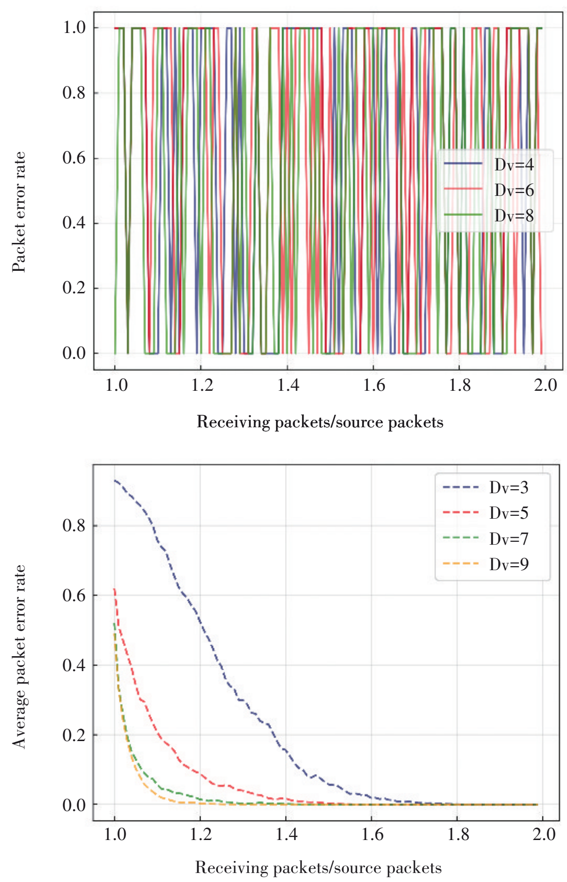
4.1 Improved Medium-Long LDPC Codes

As linear block codes, an LDPC or an XOR code is defined by its parity-check matrix $\boldsymbol{H}$ of dimensions $(n-k)\times n$, where n represents the number of all packets and k is the source packet number. The entries of the parity-check matrix $\boldsymbol{H}$ are exclusively 1 or 0, which means that it operates in the Galois Field GF(2). The parity-check matrix is so named because it provides $n-k$ parity-check equations that generate constraints between data bits and parity bits. Moreover, an LDPC code is defined as a linear block code for which the parity-check matrix $\boldsymbol{H}$ is very sparse, which means a low density (LD) of 1.

We construct an LDPC parity-check matrix $\boldsymbol{H}$ using the progressive edge growth (PEG) method, where the code length and the variable node (Dv) can be adjusted, like the physical layer LDPC channel coding. However, unlike before, when Dv is even, the decoding result fails irregularly, and the decoding result has a foreseeable change when Dv is odd, as shown in Fig. 2. Because in the binary erasure channel, the erased bits may appear to be unevenly distributed, and bits will interfere with each other when Dv is even. So only odd values of Dv can be selected. In addition, Dv represents the protection of the source packet in relation to the redundant packet. Therefore, LDPC decoding performance increases as Dv increases. At the same time, the decoding delay as a cost also increases. As shown in Fig. 3, with the number of Dv increases, there is a noticeable decrease within the range of 3 – 7, followed by a stabilizing trend. When Dv increases to a threshold value, the improvement in decoding performance no longer changes significantly as Dv increases further, but the latency still shows a linear increase (as shown in Fig. 4). Considering the delay and error correction performance, the threshold value of Dv is chosen to be 7 in the application layer of the LDPC scheme.

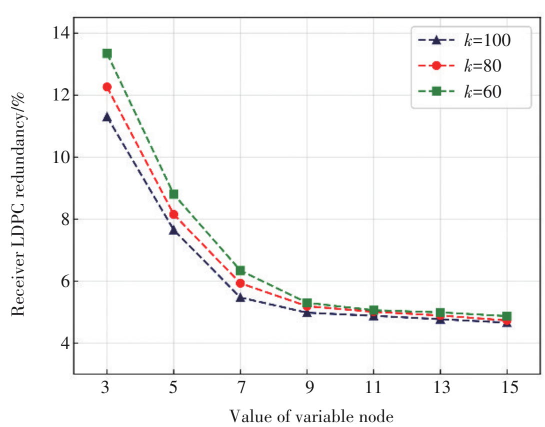
Then, to generate the code vector $\boldsymbol{c}$ from the data vector $\boldsymbol{s}$, we define the generator matrix $\boldsymbol{G}$, which holds:

$$\boldsymbol{c} = \boldsymbol{s}\boldsymbol{G}\,. \tag{1}$$



Dv: variable node

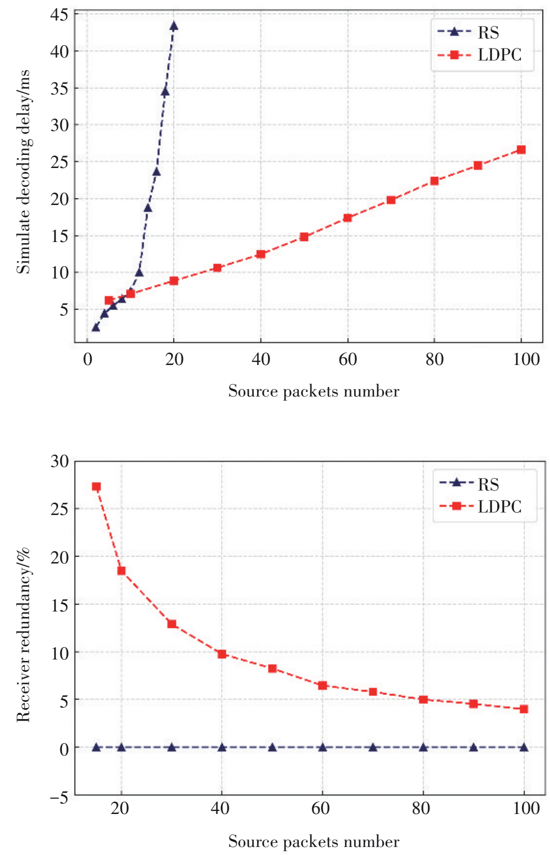
▲ **Figure 2. Packet error rate performs irregularly when Dv is even or odd**



LDPC: low-density parity-check

▲ **Figure 3. Average receiver redundancy in different values of variable nodes**

The main algorithms that create $\boldsymbol{G}$ from $\boldsymbol{H}$ consist in arranging $\boldsymbol{H}$ in an appropriate form that allows to develop $\boldsymbol{G}$ and construct it in a systematic form. Thus, $\boldsymbol{H}$ is randomly generated and then organized as:



LDPC: low-density parity-check RS: Reed-Solomon

▲ **Figure 4. RS and LDPC performance with the number of source packets**

$$\boldsymbol{H} = [\,\boldsymbol{P}^T | \boldsymbol{I}_{n-k}\,], \tag{2}$$

where $\boldsymbol{I}_{n-k}$ is the identity matrix of dimensions $(n-k)\times(n-k)$ and $\boldsymbol{P}$ is a sparse matrix of dimensions $k\times(n-k)$. So, the corresponding $\boldsymbol{G}$ matrix is:

$$\boldsymbol{G} = [\,\boldsymbol{I}_k | \boldsymbol{P}\,]. \tag{3}$$

This approach is based on the use of the Gauss-Jordan elimination. However, if we transform the right part of $\boldsymbol{H}$ to an identity matrix $\boldsymbol{I}_{n-k}$, there is no way to ensure $\boldsymbol{P}$ is a sparse matrix. So we define $\boldsymbol{H}$ as the encoding matrix $\boldsymbol{G}$ directly. The encoding matrix $\boldsymbol{H}$ of dimensions $n\times k$ is different from the parity-check matrix $\boldsymbol{H}$ before. We define the encoding vector $\boldsymbol{sp}$ as:

$$\boldsymbol{sp} = \boldsymbol{Hs}\,. \tag{4}$$

We generate code vector $\boldsymbol{sp}$ from data vector $\boldsymbol{s}$. In the system code, $\boldsymbol{sp}$ includes $\boldsymbol{s}$, which means the source bit/byte and the redundancy bit/byte are separated. Whether it is a system or non-system code, we can rebuild the coefficient matrix $\boldsymbol{H}'$

to restore the original data by the approach of the Gauss-Jordan elimination as:

$$sp' = H's , \tag{5}$$

where a certain correspondence exists between sp' and H'. sp' means the accepted sequence, and H' means the corresponding row in H with sp'. Because in the packet erasure channel, a lost bit/byte can be located at a specific position.

The decoding end (LDPC uses soft decision at the physical layer, while at the application layer, the hard decision is used. Therefore, we employ the Gaussian elimination method at the decoding end) reassembles the received packets, where the rows in the coefficient matrix H' correspond to the sp' code bits. If the reassembled matrix is full rank, it satisfies the Gaussian elimination decoding requirements:

$$H'_{m \times k} s_k = sp'_m . \tag{5.1}$$

The receiver needs to provide feedback to the sender regarding the overall packet loss rate based on the total number of packets received. Once decoding is successful, the received information packets are arranged in sequence and the video data are extracted based on the block header information. With the improvement, LDPC can perform even better in the binary erasure channel in the application layer.

4.2 Length Bounds for LDPC and RS

For encoders, the coding method directly affects the delay and effectiveness of packet loss recovery. We use a hybrid coding method of LDPC and RS, and restrict the code lengths of LDPC and RS in their own optimal interval to cover all code length requirements.

The RS code, as an ideal code, can be successfully decoded when receiving several packets equal to the number of information source packets. So, while the sender pays for extra n−k redundant packets, the receiver only needs to receive k arbitrary packets to recover the source packet. However, as the code length increases, the computational complexity of the RS code also increases sharply. In contrast, LDPC is a non-ideal code, so it is necessary to receive extra redundancy to ensure successful decoding (when the decoding sparse matrix is not full rank, the Gaussian elimination method cannot be used). However, the computational complexity of LDPC increases linearly with the code length. LPDC is therefore expected to perform better in the long code region.

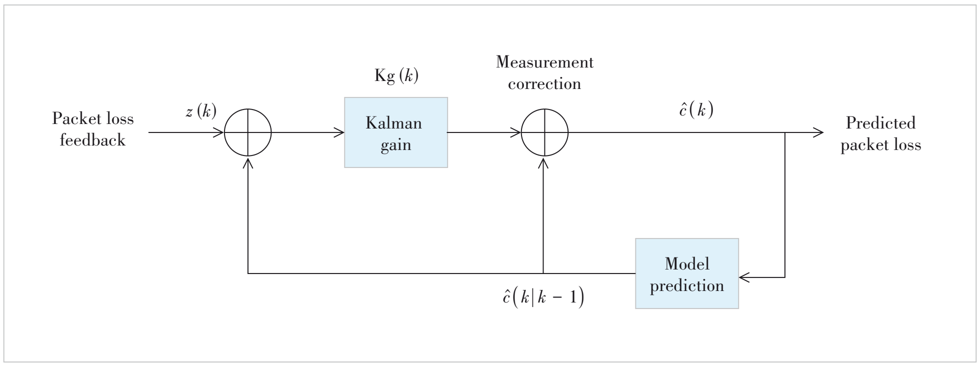
Fig. 4 shows the performance of RS and LDPC in the decoding end under the simulation environment. It is shown that RS has better performance in short codes, but as the code length increases, the delay of RS cannot meet requirements. Therefore, we choose to use LDPC instead of RS. LDPC operates in the binary field GF(2) different from RS in GF(2^8). It causes LDPC as an upper-layer coding method to have a lower decoding delay but also requires considering the additional redundancy at the receiving end of LDPC. Fig. 4 shows as the code length increases, LDPC has a high receiver redundancy in 20 source packets and it gradually becomes lower and closer to the ideal code performance, so we decide to use RS coding for code lengths not greater than 20 and LDPC coding for other situations. Besides, we set the upper limit of N_{max} comprehensive weight decoding delay and receiver redundancy to meet video transmission needs (the upper limit is relatively flexible due to LDPC characteristics, but we give an upper limit according to practical needs as shown in Section 4.1).

In Section 3, we limit the selection of RS and LDPC code lengths to certain intervals. However, these intervals are chosen to optimize the performance of RS and LDPC encoding and decoding within those specific code length ranges. It does not mean that we are restricted to only those code lengths.

5 Code Rate Adaptation Algorithm

FEC encoding redundancy allocation is mainly based on the estimated packet loss rate of the channel. Therefore, we establish a packet loss rate prediction module based on a multi-step Kalman filter system which is shown in Fig. 5. By calculating the linear minimum mean square error of the data and iteratively predicting results, we can obtain the next predicted value. The prediction process of this algorithm involves using the optimal result predicted at time $k-1$ and the measurement value at time k to calculate and update the optimal result of the state prediction at time k, and then continue iterating to obtain the predicted value of the next time.

First, we determine the state space equation used to estimate the packet loss rate of the channel. The state prediction equation is $c_k = Ac_{k-1} + Bu_k + w_k$, where A is the state transition matrix, c_{k-1} is the previous prediction value, B is the input gain matrix u_k system input vector, w_k has a mean of 0, covariance matrix $Q = E[w_{k^2}]$, and follows a normal distribution



▲Figure 5. Packet loss ratio predictor by using a Kalman filter process noise. The state measurement equation is $z_k = c_k + v_k$, where v_k has a mean of 0, covariance matrix $\boldsymbol{R} = E[v_{k^2}]$, and follows a normal distribution measurement noise.

The single-step adaptive algorithm consists of two processes, namely prediction and correction. In the prediction phase, the filter uses the estimated packet loss rate of the previous state to predict the current state. In the correction phase, the filter uses the measurement value of the current state to correct the predicted value obtained in the prediction phase, i.e., the feedback value of the packet loss rate, to obtain a new estimate value that is closer to the true value. This new estimate value is the Kalman estimate value, which is used as the estimate of the previous state for the next Kalman estimate. Since the uncertainty of process noise and measurement noise cannot be modeled, the Kalman filter is used to continuously correct the estimation model to minimize the mean square error between the true value and the estimated value. The main steps are as follows:

$$\text{Step1: } Kg(k) = \frac{P(k-1) + W}{P(k-1) + W + Q}, \tag{6}$$

$$\text{Step2: } \hat{c}(k) = \hat{c}(k|k-1) + Kg(k)\left[z(k) - \hat{c}(k|k-1)\right], \tag{7}$$

$$\text{Step3: } P(k) = \left(1 - Kg(k)\right)\left(P(k-1) + W\right), \tag{8}$$

where $P(k) = E[(\hat{c}(k) - c(k))^2]$ is the error variance of the model. Through the Kalman filter, we obtain a packet loss rate c_k at a certain time scale.

To obtain the multi-step packet loss rate prediction value, the average packet loss rate of five blocks in the future is predicted at each step. Since the average packet loss rate measurement $z(k+1), z(k+2), z(k+3), z(k+4), z(k+5)$ is still unknown when encoding block $z(k)$ of the next blocks meets $z(k+1) = z(k+2) = z(k+3) = z(k+4) = z(k+5) = z(k)$.

As an important parameter for adaptive redundancy calculation, the packet loss is accompanied by another important parameter, which is the redundancy at the receiving end. Given that the RS code is ideal, decoding can be successfully achieved by receiving any k packets. However, the long-code LDPC method is not an ideal one, and the scheme needs to establish the cost of receiving redundancy at the receiver, which describes the mapping relationship between the number of additional packets required by the receiver and the retransmission rate due to decoding failures.

We establish a receiver redundancy cost function, and the selected code rate is influenced by the receiver redundancy cost. The goal is to minimize the overall transmission cost as much as possible. The receiver redundancy cost of the target video can be characterized by the receiver redundancy cost function:

$$C = [Q(k, r-m) - \delta \cdot P_{\mathrm{rtt}}(k, r-m)]^2, \tag{9}$$

where C represents the receiver redundancy cost of the t-th block, i.e., the total cost of transmitting block t at the current code rate. $Q(k, k+r-m)$ represents the code rate with the source information bit length of k and the redundancy bit length of r (since the medium-long code is not in the form of a system code, r can be expressed as $n-k$, where n is the total information bit length after encoding), and m represents the number of lost packets. The code rate can be expressed as $k/(k+r)$ and the packet loss rate can be expressed as $m/(k+r)$. $k+r-m$ represents the number of packets received, and $Q(k, r-m)$ represents the function related to k and $k+r-m$. Optionally, $Q(k, r-m) = (r-m)/k$ can be directly expressed as the redundancy ratio of the receiver, where $P_{\mathrm{rtt}}(k, r-m)$ represents the decoding failure rate under the current receiver redundancy ratio which is the ratio of the overall decoding errors of the block in the simulation process. Decoding failure requires retransmission of the block, and δ represents the weight of the decoding failure item. An increasing value of δ indicates that we consider the current decoding failure rate to be unacceptable.

By reversely solving the redundancy cost function problem, we can obtain the proportion of additional packets that the receiver needs to receive, r_{rec}, when the expected retransmission probability is not greater than a probability P using LDPC encoding. Then, by estimating the packet loss rate below, we can obtain the encoding redundancy:

$$r = \left[\left(k + \frac{k}{r_{\mathrm{rec}}}\right)/c\right] - k, \tag{10}$$

where c is the current packet loss estimate and k is the length of the information source code.

6 Evaluation

6.1 Setup

1) Datasets: This paper considers a transmission channel based on the Gilbert-Elliott model[18], which is acknowledged as a common simulation environment of network packet delivery. The Gilbert-Elliott model is a Markov process, where B and G indicate the bad and good network state. The probability of transitioning from state B to G is denoted as P_{BG}, and the probability of transitioning from state G to B is denoted as P_{GB}. The transition matrix of the Markov chain is as follows:

$$\boldsymbol{A} = \begin{pmatrix} 1 - P_{\mathrm{GB}} & P_{\mathrm{GB}} \\ P_{\mathrm{GB}} & 1 - P_{\mathrm{GB}} \end{pmatrix}. \tag{11}$$

In a stable state, π_{G} is the probability of being in a good state and π_{B} a bad state.

$$\pi_{\mathrm{G}} = \frac{P_{\mathrm{GB}}}{P_{\mathrm{GB}} + P_{\mathrm{GB}}},$$

$$\pi_{\mathrm{B}} = \frac{P_{\mathrm{GB}}}{P_{\mathrm{GB}} + P_{GB}}. \tag{12}$$

The formula for calculating the probability of packet loss is:

$$P_E = \pi_{\mathrm{G}}(1 - k) + \pi_{\mathrm{B}}(1 - h), \tag{13}$$

where k represents the probability of successful reception in a good state, and h represents the probability of packet loss in a bad state. Therefore, P_E is decided by setting the value of four parameters. Table 1 shows four parameter values for network packets loss ratio range of 5% – 20%, 20% – 40%, and 40% – 80%.

2) The high-bitrate 4K/30fps video is performed frame-level cutting by FFMPEG. We sample the generalized GE channel with different packet loss rates based on the total number of blocks. Then, the proposed scheme is compared with the WebRTC FEC algorithm in the GE channel:

• WebRTC-FEC based on the XOR algorithm employs a redundancy protection scheme. When the original packet size is less than or equal to 12, the redundancy level is directly obtained from a lookup table and the packet is encoded. When the original packet size is greater than 12, an interval-based grouping redundancy encoding method is used. The adaptive solution of WebRTC predicts the network status based on the video bitrate and the packet loss rate of video transmission. The redundancy level is obtained from a lookup table based on the video bitrate and packet loss rate, and is combined with the network status prediction that increases the RTT of transmission.

• The proposed AH-FEC is a long and short code adaptive redundancy coding algorithm based on RS and LDPC codes. It selects the coding redundancy degree based on the decoding redundancy at the receiver and the packet loss rate of the channel, balancing system latency and redundancy. At the encoding end, the future packet loss rates of multiple video blocks are predicted based on the packet loss rate of past video blocks for a certain period. Then, the encoding redundancy is dynamically adjusted accordingly.

3) Metric: In performance comparison, we consider the following measurement metrics.

• Data recovery ratio. The percentage of data blocks that are successfully recovered is the proportion of all data blocks.

• Redundancy ratio. The redundancy ratio is the proportion of encoded extra packets relative to packets, and it is expressed as $\frac{n-k}{k}$.

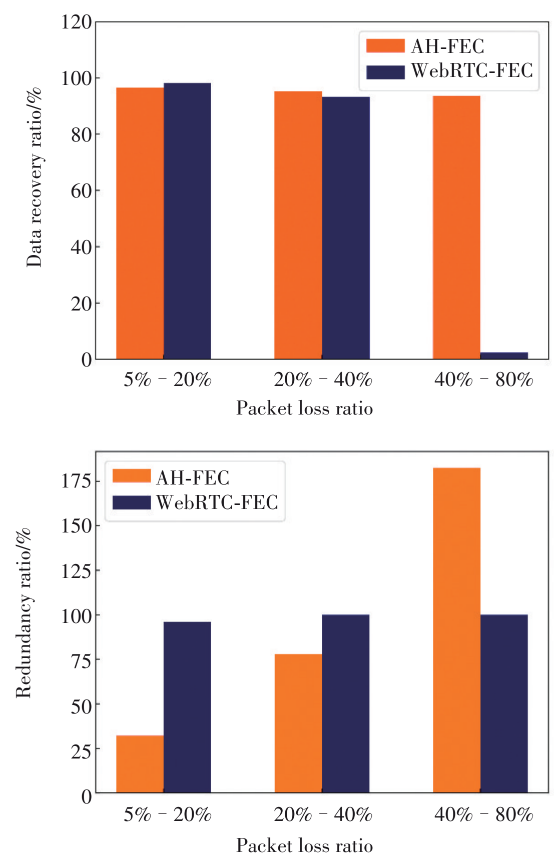
▼Table 1. Parameter values for different packets loss ratio ranges

Parameter	$P_{\mathrm{GB}}(p)$	$P_{\mathrm{BG}}(r)$	k	h	P_E	Range
Value	0.130	0.910	0.970	0.030	0.114	(0.050, 0.200)
	0.360	0.840	0.980	0.050	0.299	(0.200, 0.400)
	0.900	0.600	0.980	0.020	0.596	(0.400, 0.800)

6.2 Experiments on Simulation

Simulation traces contain 300 sample frames, and the redundancy strategy in WebRTC treats each frame as a block directly. We reorganize it according to the frame partition method and evaluate the results of the two FEC schemes.

Fig. 6 presents the recovery ratio and redundancy ratio of the two algorithms under network packet loss rates of 5% – 20%, 20% – 40%, and 40% – 80%. As evidenced by Fig. 6, the recovery success rates of both the proposed scheme and the WebRTC scheme approach 100% at packet loss rates of 5% to 20%, while the proposed scheme only requires about 32% redundancy. This lower redundancy requirement is attributed to the proposed scheme using RS-LDPC hybrid coding, which proves more efficient than the XOR coding employed by WebRTC, thereby requiring less redundancy to offer comparable protection capability. In addition, the adaptive capability of Kalman filtering is also superior to the fixed redundancy table of WebRTC, so the redundancy rate is greatly compressed. When the redundancy rate ranges between 20% and 40%, the WebRTC redundancy reaches its peak in the redundancy table at 100%. At this stage, a substantial reduction in the data recovery rate is observed due to WebRTC's lim-



AH-FEC: adaptive hybrid forward error correction

▲Figure 6. Data recovery and redundancy ratio in different packet loss ratios

ited capability. In contrast, the proposed solution outperforms WebRTC in terms of both recovery and redundancy rates. Finally, the proposed solution works stably even at packet loss rates of 40% to 80%, when the redundancy rate reaches over 180% and the recovery rate reaches over 93%. In contrast, the redundancy of WebRTC is limited by an offline table so the FEC cannot work.

To illustrate the specific results in Fig. 6, we present the specific results of data recovery ratio and redundancy ratio in Table 2.

In summary, the proposed scheme presents the following advantages over the current FEC techniques (RFC5109) implemented in WebRTC:

- For network packet loss rates of 5% – 20%, the retransmission rate is kept at a relatively low level, achieving a 65.65% reduction in the redundancy ratio of the sent data.
- For network packet loss rates of 20% – 40%, in comparison to WebRTC, this approach improves the data recovery ratio by 2.21% and decreases redundant data by 22.06%.
- For network packet loss rates of 40% – 80%, the redundancy ratio increases by 82.56%, achieving a tremendous reduction in the retransmission rate.

These results suggest that the redundancy strategy employed by WebRTC lacks adaptability and depends heavily on retransmission, making it unsuitable for high-bitrate videos. Conversely, the proposed AH-FEC succeeds in reducing both redundancy and retransmission rates under identical conditions, demonstrating adaptability to complex channel loss scenarios, especially in weak network environments with high packet loss rates.

7 Conclusions

The FEC scheme employed in WebRTC is unsuitable for high-bitrate video due to limitations in coding efficiency and adaptive capacity. Therefore, we develop a hybrid coding method based on RS/LDPC codes, which determines the coding redundancy according to the receiver redundancy and packet loss rate. When contrasted with the group XOR method in WebRTC, the proposed scheme significantly reduces sending redundancy while ensuring low delay and high recovery rate. We also implement a redundancy decision algorithm based on multi-step packet loss rate prediction, which generates forward-looking redundancy decisions based on feedback from the packet loss rate of the receiver. In comparison to the static table lookup method in WebRTC, this approach can adapt to complex and dynamic packet loss environments. The proposed AH-FEC consistently maintains a high data recovery ratio with the interval change of packet loss rates.

▼Table 2. Performance comparison of two adaptive FEC methods in different packet loss rates

(a) Packet loss ratio: 5% – 20%

	WebRTC	AH-FEC
Data recovery ratio	98.20%	96.49%
Redundancy ratio	95.89%	32.18%

(b) Packet loss ratio: 20% – 40%

	WebRTC	AH-FEC
Data recovery ratio	2.29%	93.54%
Redundancy ratio	100.00%	182.56%

(c) Packet loss ratio: 40% – 80%

	WebRTC	AH-FEC
Data recovery ratio	93.23%	95.29%
Redundancy ratio	100.00%	77.94%

AH-FEC: adaptive hybrid forward error correction

References

[1] CAI S H, ZHAO S C, MA X. Free ride on LDPC coded transmission [J]. IEEE transactions on information theory, 2022, 68(1): 80 – 92. DOI: 10.1109/TIT.2021.3122342

[2] JUN M. Binary polar codes based on bit error probability [C]//Proceedings of IEEE International Symposium on Information Theory (ISIT). IEEE, 2022: 2148 – 2153. DOI: 10.1109/ISIT50566.2022.9834407

[3] XU J L, CHEN W, AI B. Deep Joint source-channel coding based CSI feedback [J]. ZTE Technology journal, 2022, 27(2): 29 – 33. DOI: 10.12142/ZTETJ.202302007

[4] KOTABA R, MANCHÓN C N, BALERCIA T, et al. How URLLC can benefit from NOMA-based retransmissions [J]. IEEE transactions on wireless communications, 2021, 20(3): 1684 – 1699. DOI: 10.1109/TWC.2020.3035517

[5] WANG Y, ZHU Q F. Error control and concealment for video communication: a review [J]. Proceedings of the IEEE, 1998, 86(5): 974 – 997. DOI: 10.1109/5.664283

[6] LI A. RTP Payload Format for generic forward error correction [EB/OL]. [2023-07-10]. https://www.rfc-editor.org/info/rfc5109

[7] MACKAY D J C. Fountain codes [J]. IEE proceedings-communications, 2005, 152(6): 1062 – 1068. DOI: 10.1049/ip-com: 20050237

[8] DEMIR U, AKTAS O. Raptor versus Reed-Solomon forward error correction codes [C]//International Symposium on Computer Networks. IEEE, 2006: 264 – 269. DOI: 10.1109/ISCN.2006.1662545

[9] BOURAS C, KANAKIS N, KOKKINOS V, et al. Evaluating RaptorQ FEC over 3GPP multicast services [C]//The 8th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2012: 257 – 262. DOI: 10.1109/IWCMC.2012.6314213

[10] WICKER S B, BHARGAVA V K. Reed-Solomon codes and their applications [M]. New York: John Wiley & Sons, 1999

[11] SUDAN M. Decoding of reed Solomon codes beyond the error-correction bound [J]. Journal of complexity, 1997, 13(1): 180 – 193. DOI: 10.1006/jcom.1997.0439

[12] ATIYA A E, YOO S G, CHONG K T, et al. Packet loss rate prediction using the sparse basis prediction model [J]. IEEE transactions on neural networks, 2007, 18(3): 950 – 954. DOI: 10.1109/TNN.2007.891681

[13] EMARA S, FONG S L, LI B C, et al. Low-latency network-adaptive error control for interactive streaming [C]//Proceedings of IEEE Transactions on Multimedia. IEEE, 2021: 1691 – 1706. DOI: 10.1109/TMM.2021.3070134

[14] CHENG S, HU H, ZHANG X G, et al. DeepRS: Deep-learning based network-adaptive FEC for real-time video communications [C]//Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020: 1 – 5. DOI: 10.1109/ISCAS45731.2020.9180974

[15] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735 – 1780. DOI: 10.1162/

neco.1997.9.8.1735
[16] HU H, CHENG S, ZHANG X G, et al. LightFEC: network adaptive FEC with a lightweight deep-learning approach [C]//The 29th ACM International Conference on Multimedia. ACM, 2021: 3592 – 3600. DOI: 10.1145/3474085.3475528
[17] WANG R, SI L, HE B F. Sliding-window forward error correction based on reference order for real-time video streaming [J]. IEEE access, 2022, 10: 34288 – 34295. DOI: 10.1109/ACCESS.2022.3162217
[18] HASSLINGER G, HOHLFELD O. The gilbert-elliott model for packet loss in real time services on the Internet [C]//The 14th GI/ITG Conference: Measurement, Modelling and Evaluations of Computer and Communication Systems. IEEE, 2008: 1 – 15

Biographies

XIONG Yuhui is pursuing his master degree at the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering and School of Electronic Information and Communications, Huazhong University of Science and Technology, China. His research interests include multimedia transmission technology and cell-free massive MIMO.

LIU Zhilong is currently the cloud video system chief planning engineer of ZTE Corporation. His main research directions are multimedia transmission technology, SRTN products, cloud desktop products and remote secure office solutions.

XU Lingmin is currently the cloud video product chief planning engineer of ZTE Corporation. His main research directions are cloud computing, IP-based multimedia transmission technology, IPTV/OTT products, cloud desktop products and remote secure office solutions.

HUA Xinhai is currently the Vice President of ZTE Corporation and general manager of the cloud video product project. His main research directions are cloud computing, IP-based video product technology and solutions, security solutions of video service, technology and product solutions of content distribution network, etc.

WANG Zhaoyang is pursuing his PhD degree at the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering and School of Electronic Information and Communications, Huazhong University of Science and Technology, China. His research interests include multimedia transmission technology and cell-free massive MIMO.

BI Ting (ting.bi@ieee.org) is currently an associate professor with the Research Center of 6G Mobile Communications and School of Cyber Science and Engineering, Huazhong University of Science and Technology, China. He received the BE degree in software engineering from Wuhan University, China in 2010, and the ME and PhD degrees in telecommunications from Dublin City University, Ireland in 2011 and 2017, respectively. His research interests include mobile and wireless communications, multimedia and multi-sensory media streaming.

JIANG Tao is currently a Distinguished Professor with the Research Center of 6G Mobile Communications and School of Cyber Science and Engineering, Huazhong University of Science and Technology, China. He received a PhD degree in information and communication engineering from Huazhong University of Science and Technology in 2004. He is/was a symposium technical program committee membership of some major IEEE conferences, including INFOCOM, GLOBECOM, and ICC. He was invited to serve as a TPC Chair for IEEE GLOBECOM 2013, IEEE WCNC 2013, and ICCC 2013. He is/was an associate editor of some technical journals in communications, including the *IEEE Network*, *IEEE Transactions on Signal Processing*, *IEEE Communications Surveys and Tutorials*, *IEEE Transactions on Vehicular Technology*, and he is the area editor of *IEEE Internet of Things Journal* and associate editor-in-chief of *China Communications*.