EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Optimized cross-layer forward error correction coding for H.264 AVC video transmission over wireless channels

Ali Talari[1]*, Sunil Kumar[2], Nazanin Rahnavard[1], Seethal Paluri[2] and John D Matyjas[3]

Abstract

Forward error correction (FEC) codes that can provide *unequal error protection* (UEP) have been used recently for video transmission over wireless channels. These video transmission schemes may also benefit from the use of FEC codes both at the *application layer* (AL) and the *physical layer* (PL). However, the interaction and optimal setup of UEP FEC codes at the AL and the PL have not been previously investigated. In this paper, we study the cross-layer design of FEC codes at both layers for H.264 video transmission over wireless channels. In our scheme, UEP *Luby transform codes* are employed at the AL and *rate-compatible punctured convolutional* codes at the PL. In the proposed scheme, video slices are first prioritized based on their contribution to video quality. Next, we investigate the four combinations of cross-layer FEC schemes at both layers and concurrently optimize their parameters to minimize the video distortion and maximize the peak signal-to-noise ratio. We evaluate the performance of these schemes on four test H.264 video streams and show the superiority of optimized cross-layer FEC design.

Keywords: Video transmission; Forward error correction; LT codes; RCPC codes; Cross-layer; Wireless channels; Video coding; H.264

1 Introduction

Multimedia applications such as video streaming, which are delay sensitive and bandwidth intensive, are growing rapidly over wireless networks. However, existing wireless networks provide only limited bandwidth and time-varying *quality of service* (QoS) support for these applications. Due to limited wireless bandwidth, the video is compressed using sophisticated compression techniques such as H.264 AVC, which is the state-of-the-art video compression standard jointly developed by the ITU and ISO [1]. The compressed video is vulnerable to channel impairments as the corrupted packets induce different levels of quality degradation due to temporal and spatial dependencies in the compressed bitstream. The most important problem that affects video quality is error propagation where an error in a reference frame is propagated by the decoder to all future reconstructed frames, which are predicted from the corrupted reference frame. This problem has led to the design of error-resiliency features, such as *flexible macroblock ordering* (FMO) [2], data partitioning, and error concealment schemes in H.264 [1,3,4]. Recent research has demonstrated the promise of cross-layer protocols for supporting the QoS demands of multimedia applications over wireless networks [5-7]. For example, van der Schaar and Shankar [6] showed the benefits of the joint APP-MAC-PHY approach for transmitting video over wireless networks.

Forward error correction (FEC) schemes are used to protect the video data against channel errors in order to improve the successful data transmission probability and to eliminate the costly retransmissions. However, the maximum throughput does not guarantee the minimum video distortion at the receiver for the following reasons. *First*, unlike data packets, loss of H.264 compressed video slices induces different amounts of distortion in the received video. Therefore, the FEC code rates should be adaptive to the slice priority. *Second*, video data are delay sensitive; therefore, the retransmission of corrupted slices may not be feasible. *Third*, a video stream can tolerate

*Correspondence: ali.talari@okstate.edu
[1]Department of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA
Full list of author information is available at the end of the article

loss of some slices because the lost slices can be error-concealed. This is true especially for the low-priority slices, which introduce low distortion in the received video and result in graceful quality degradation. In this paper, we consider H.264 AVC streams with fixed slice sizes, where each slice can be independently decoded. The video slices are classified into *four* priority classes based on the distortion contributed by their loss to the received video quality.

An FEC code that provides *unequal error protection* (UEP), i.e., a higher (lower) protection to high (low)-priority video slices, can achieve considerable quality improvement compared to the *equal error protection* (EEP) FEC codes [8,9]. Note that the UEP FEC codes may be employed both at the *application layer* (AL) and *physical layer* (PL). Recently, some schemes [5,10,11] have considered the precise tuning of EEP FEC schemes at the AL and the PL. However, to the best of our knowledge, existing schemes have not investigated the cross-layer design of *UEP* FEC codes at the AL and the PL for prioritized video transmission. Employing FEC codes at both layers introduces two interesting trade-offs that we investigate in this paper. *First*, both FEC codes share a common channel bandwidth to add their redundancy and the optimal ratio of overhead added by each needs to be determined for a given channel signal-to-noise ratio (SNR) and bandwidth. *Second*, since UEP can be provided at both layers, we need to find the optimal UEP/EEP FEC setup to maximize the video peak SNR (PSNR). To tackle these trade-offs, we *concurrently* tune the parameters of two FEC codes at both layers.

We use *UEP Luby transform (LT)* codes [12,13] at the AL and *rate-compatible punctured convolutional* (RCPC) codes [14] at the PL. LT codes [15] are modern and efficient FEC codes that are specifically suitable for packet-level coding at the AL. These codes are *rateless* [12,13,15,16] in the sense that they can generate unlimited encoded information from a finite-length source information.

Next, we carry out a cross-layer optimization to find the optimal parameters of both FEC codes by considering the relative priorities of video packets. For a known channel SNR (i.e., $\frac{E_s}{N_0}$), we address the problem of assigning optimal FEC code rates at the AL and the PL to the individual priority slices within the channel bit-rate limitations. The information about the channel conditions can be obtained from the receiver in the form of channel side information [5-7,17,18].

The scheme provides higher transmission reliability to high-priority slices at the expense of the higher loss rates for low-priority slices and, whenever necessary, also discards some low-priority slices to meet the channel bit-rate limitations. We show that adapting the FEC code rates to the slice priority reduces the overall expected video distortion at the receiver. Our scheme does not assume retransmission of lost slices. The preliminary results of this paper appeared in [8].

This paper is organized as follows: Section 2 provides an overview of the related work on FEC coding for video streams. Section 3 provides a brief background on the LT and RCPC FEC codes. Section 4 describes the video slice priority assignment, design of LT and RCPC codes, and cross-layer FEC schemes. Section 5 presents the cross-layer optimization and performance of the proposed FEC schemes. The simulation results of the proposed cross-layer FEC schemes on sample H.264 videos are presented in Section 6, followed by conclusions in Section 7.

2 Related work

LT codes have recently become popular in video transmission schemes due to their good performance and low complexity [15]. Kushwaha et al. [19] used LT codes to encode *group of pictures* (GOP) of each layer of H.264 SVC video for transmission over cognitive radio wireless networks. Ahmad et al. [17] took advantage of the ratelessness of LT codes and proposed an adaptive FEC scheme for video transmission over the Internet by employing feedback from receivers in the form of acknowledgement. Cataldi et al. [18] proposed a novel LT code, called sliding-window Raptor codes, with a higher efficiency than regular LT codes. They used these codes to provide UEP for a two-layer H.264 SVC scalable video. LT codes were also used in [20-25] to design streaming schemes with lower complexity.

Stockhammer et al. [5] defined the protocol stack, including the FEC coding at the AL and the PL, for the *multimedia broadcast multicast service* (MBMS) download and streaming in universal mobile telecommunication system (UMTS). In [5], a Raptor code [16] is used at the AL and a turbo code at the PL. Gomez-Barquero and Bria [10] suggested employing the Raptor codes as the AL FEC in DVB-H systems for mobile terminals and demonstrated its advantages over conventional *multiprotocol encapsulation* (MPE) FEC. Conventional MPE FEC employs the Reed-Solomon codes to encode the video stream; hence, it lacks the flexibility of LT coding at the AL. Courtade and Wesel [11] considered a setup with LT coding at the AL and turbo coding at the PL, and showed that the available channel bandwidth should be optimally split between the AL and PL FEC codes to improve the system performance.

Luby et al. [26] also considered employing two layers of EEP FEC at the AL and the PL for MBMS download delivery in UMTS. They investigated the trade-off between the AL FEC and PL FEC codes, and studied the advantages of the AL FEC on the system performance. Stockhammer and Liebl [27] used the Raptor codes at the AL in 3GPP streaming applications. They investigated how the AL

FEC coding may guarantee the ratio of satisfied users who are receiving the video stream. Afzal et al. [28] investigated the overall system performance when the AL FEC codes are used in video streaming in UMTS and packet radio services. Alexiou et al. [29] studied the power control of streaming over *high-speed downlink packet access* systems when the AL FEC is employed. Munaretto et al. [30] proposed an interesting optimization of the AL FEC coding, video source coding, and the PL rate selection to improve the PSNR of delivered video on cellular networks. The authors in [31] also considered employing the Raptor codes at the AL to improve the quality of service for video in MBMS in *long-term evolution (LTE)* networks. They investigated the benefits of the AL FEC to multicast multimedia contents and examined how much FEC redundancy should be used under different packet loss patterns.

In [8], we investigated UEP rateless coding at the AL and assumed an ideal PL coding. We found the optimal parameters of a UEP rateless code that maximizes the video quality at the receiver for known channel bandwidth. In this paper, we extend the results of [8] and consider the interaction of the AL coding with the PL coding in video transmission schemes.

3 Background

In this section, we briefly review LT and RCPC FEC codes that will be used at the AL and the PL, respectively, in our proposed cross-layer FEC scheme.

3.1 LT codes

Recently, a new class of FEC codes called rateless (Fountain) codes has been invented. LT codes [15] and Raptor codes [16] are examples of such codes. Unlike other FEC codes, such as LDPC codes [32], rateless codes can adapt to any erasure channel with unknown or varying characteristics as they do not impose any code rate constraint. Fountain codes are especially very desirable for packet-level coding at the application layer, where the underlying channel can be modeled as a packet erasure channel.

LT codes can generate a limitless number of *output symbols* from N_s *input symbols* based on a degree distribution $\{\Omega_1, \Omega_2, \ldots, \Omega_{N_s}\}$, where Ω_i is the probability that an output symbol has degree i, and $\sum_{i=1}^{N_s} \Omega_i = 1$. This probability distribution can also be shown by its generator polynomial $\Omega(x) = \sum_{i=1}^{N_s} \Omega_i x^i$. In LT coding, first an output symbol degree d is randomly chosen from $\Omega(.)$. Next, d input symbols are chosen *uniformly* and *randomly* from N_s input symbols and are bit-wise *XOR*ed together to generate an output symbol. $\Omega(.)$ is usually fine-tuned such that the N_s input symbols can be decoded from any $\gamma_r N_s$ output symbols, for γ_r slightly greater than 1. Here, γ_r is the *received coding overhead.* LT decoding is performed *iteratively.* At each iteration, an output symbol is found such that the value of all but one of its neighboring input symbols is known. The value of the unknown input symbol is computed by a simple XOR. This step is applied iteratively until no more such output symbols can be found.

Robust-Soliton degree distribution was designed by Luby for LT codes [15]. LT coding with Robust-Soliton distribution results in asymptotically capacity-achieving codes with the encoding complexity of $O(N_s \log N_s)$. To reduce the coding complexity to linear (at the cost of a slight performance loss), new degree distributions for LT codes have been introduced such as [16]

$$\begin{aligned}\Omega(x) = {} & 0.00797x + 0.49357x^2 + 0.16622x^3 \\ & + 0.07265x^4 + 0.08256x^5 \\ & + 0.05606x^8 + 0.03723x^9 + 0.05559x^{19} \\ & + 0.02502x^{65} + 0.00314x^{66}.\end{aligned} \tag{1}$$

In this paper, we use (1) as the degree distribution of LT codes.

Interestingly, it has been shown that LT codes can easily provide UEP property with a slight change in the encoding process. In [12,13], the authors proposed UEP LT codes by modifying the source symbol selection from uniform to *non-uniform.* In UEP LT codes, N_s source symbols are partitioned into r sets, $s_1, s_2, \ldots, s_r$ of sizes $\tau_1 N_s, \tau_2 N_s, \ldots, \tau_r N_s$, such that $\sum_{j=1}^{r} \tau_j = 1$. Let p_j be the probability that a source symbol from set s_j is chosen to form an encoded symbol. Consequently, we define the *protection level* of priority i group as $k_i = p_i N_s$, where $\sum_{j=1}^{r} k_j \tau_j = 1$. Further, let $y_{l,j}$ be the probability that a source symbol in s_j is not recovered after l LT decoding iterations at the receiver. For $j = 1, \ldots, r$ we have [12,13]

$$y_{l,j} = \delta_j(1 - \beta(1 - \sum_{m=1}^{r} p_m \tau_m N_s y_{l-1,m})), l \geq 1, \tag{2}$$

where $y_{0,j} = 1$, $\beta(x) = \Omega'(x)/\Omega'(1)$, and $\delta_j(x) = e^{N_s p_j \Omega'(1) \gamma_r (x-1)}$.

It can be shown that sequences $\{y_{l,j}\}_l, \forall j$ converge to a fixed point y_j [12,13], where y_j is the final decoding error rate of symbols in set $j \in \{1, 2, \ldots, r\}$ for a UEP LT code with the parameters $\{\Omega(x), \gamma_r, \tau_1, \tau_2, \ldots, \tau_r, p_1, p_2, \ldots, p_r\}$. For EEP LT coding, we have $k_j = 1, j \in \{1, 2, \ldots, r\}$; hence, $\forall j \in \{1, 2, \ldots, r\}, y_j = y$. Note that (2) has been derived from tree-graph approximation of LT codes and provides y_j's for asymptotic case ($N_s \to \infty$) [12,13,16].

3.2 RCPC codes

We choose RCPC codes [14] due to their flexibility in providing various code rates. RCPC codes use a *low-rate* convolutional mother code and employ various puncturing patterns to obtain various code rates. The RCPC decoder employs a *Viterbi* decoder. The bit error rate P_b of the Viterbi decoder is upper bounded by [14]

$$P_b \leq \frac{1}{P} \sum_{d=d_f}^{\infty} c_d P_d, \tag{3}$$

where d_f is the free distance of the convolutional code, P is the puncturing period, and c_d is the total number of error bits produced by the incorrect paths and is known as the *distance spectrum* [14]. Finally, P_d is the probability of selecting a wrong path in Viterbi decoding with Hamming distance d, which depends on the modulation and channel characteristics. For an RCPC code with rate R, using the additive white Gaussian noise (AWGN) channel, binary phase shift keying (BPSK) modulation, and the symbol to noise power ratio $\frac{E_S}{N_0} = R\frac{E_b}{N_0}$, the value of P_d (using soft Viterbi decoding) is given by [14]

$$P_d = \frac{1}{2}\mathrm{erfc}\sqrt{d\frac{E_S}{N_0}} = Q\left(\sqrt{2d\frac{E_S}{N_0}}\right), \tag{4}$$

where $Q(\lambda) = \frac{1}{\sqrt{2\pi}} \int_{\lambda}^{\infty} e^{-\frac{a^2}{2}} da.$

4 Cross-layer FEC coding for H.264 video bitstream

In this section, we discuss a priority assignment scheme for H.264 AVC video slices, design of LT and RCPC codes, and our proposed cross-layer FEC scheme. We consider a unicast video transmission from a source node (at the transmitter) to a destination node (at the receiver) in a single-hop wireless network and ignore the intermediate network layers, i.e., transport layer (TL), network layer (NL), and link layer (LL). This allows our algorithm to be employed with different existing network protocols stacks.

4.1 Priority assignment for H.264 video slices

In H.264 AVC, the video frames are grouped into GOPs, and each GOP is encoded as a unit. For the sake of simplicity, we use a GOP length of 30 frames which corresponds to a duration of 1 s. We encode each GOP independently by employing FEC codes. We have used a fixed slice size configuration where macroblocks of a frame are aggregated to form a fixed slice size. Let N_s be the *average* number of slices in 1 s of the video. More details of the video encoding parameters are given in Section 6.

H.264 slices can be prioritized based on their *distortion* contribution to the received video quality [9,33-37]. In this paper, the total distortion of a slice loss is computed using the *cumulative mean square error* (CMSE), which takes into consideration the error propagation within the entire GOP [9,34]. Let the original uncompressed video frame at time t be $f(t)$, the decoded frame without the slice loss be $\hat{f}(t)$, and the decoded frame with the slice loss be $\tilde{f}(t)$. Assuming that each frame consists of $N \times M$ pixels, the MSE introduced by the loss of a slice in the video frame is computed by

$$\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[(\text{pixel value}_{i,j})_{\hat{f}(t)} - (\text{pixel value}_{i,j})_{\tilde{f}(t)}\right]^2.$$

The loss of a slice in a reference frame can also introduce error propagation in the current and subsequent frames until the end of GOP. The CMSE contributed by the loss of the slice is thus computed as the sum of MSE over the current and all the subsequent frames in the GOP. Note that computation of slice CMSE requires decoding of the entire GOP for every slice loss, which introduces computational overhead. This overhead can be avoided by predicting the slice CMSE using a low-complexity scheme recently proposed by us in [9]. This slice CMSE prediction scheme uses certain parameters from the current encoded frame alone without using the future frames in the GOP.

We use the CMSE metric to determine the slice priority. All slices in a GOP are distributed into $r = 4$ priority classes of *equal size* based on their CMSE value. The priority 1 slices induce the highest distortion whereas the priority 4 slices induce the least distortion to received video quality. Note that using more than four slice priorities would result in a more accurate and flexible UEP coding at the cost of higher complexity due to a larger number of design parameters. In fact, using N_s priority levels would achieve the best performance where each slice is separately protected based on its CMSE. On the other hand, using fewer than four priority levels would limit the flexibility of our scheme and hence decrease its performance.

Let CMSE_i denote the *average* CMSE of all slices in a priority class i. Therefore, we have $\text{CMSE}_1 > \text{CMSE}_2 > \text{CMSE}_3 > \text{CMSE}_4$. Since CMSE_i may vary considerably for various videos depending on their content, we use the *normalized* CMSE_i, $\overline{\text{CMSE}}_i = \frac{\text{CMSE}_i}{\sum_{j=1}^{4}\text{CMSE}_j}$ to represent the relative importance of a priority class. We show $\overline{\text{CMSE}}_i$ for six H.264 test video sequences in Table 1. These video sequences have widely different spatial and temporal content.

Table 1 shows that the first five videos, which have very different characteristics (such as slow, moderate, and high motion), have almost similar $\overline{\text{CMSE}}_i$ values. We also observed similar $\overline{\text{CMSE}}_i$ values for other video sequences, such as Table Tennis and Mother Daughter. However, Akiyo, which is a static sequence, has different $\overline{\text{CMSE}}_i$ values than other sequences. The $\overline{\text{CMSE}}_i$ values changed only slightly when these videos were encoded at different bit rates (i.e., 512 kbps and 1 Mbps) and slice sizes (150 to 900 bytes). When these videos are encoded at 840 kbps with 150-byte slices, we get $N_s \approx 700$.

Table 1 Normalized CMSE, $\overline{\text{CMSE}}_i$, for slices in different priorities of sample videos

Sequence	$\overline{\text{CMSE}}_1$	$\overline{\text{CMSE}}_2$	$\overline{\text{CMSE}}_3$	$\overline{\text{CMSE}}_4$
Coastguard	0.61	0.22	0.12	0.05
Foreman	0.63	0.21	0.11	0.05
Bus	0.64	0.21	0.10	0.04
Football	0.65	0.21	0.10	0.04
Silent	0.68	0.2	0.09	0.03
Akiyo	0.85	0.12	0.03	0.01

We choose the $\overline{\text{CMSE}}_i$ values of Bus, which are similar to most other videos discussed above, to tune our proposed cross-layer scheme for all videos in Section 5. Since the $\overline{\text{CMSE}}_i$ values of Akiyo are different, we also study the performance of the proposed cross-layer FEC scheme for Akiyo by using its own $\overline{\text{CMSE}}_i$ values and compare it to the performance of the scheme designed using the $\overline{\text{CMSE}}_i$ values of Bus in Section 6.

4.2 Design of LT codes at the AL

The video slices may be either directly passed to the PL or encoded using an EEP/UEP LT code before passing to the PL. Therefore, the AL frames contain either uncoded or LT-coded video slices. When no LT coding is performed at the AL, each video slice forms an AL frame and the N_s AL frames are given to the lower network layers. When the LT coding is performed at the AL, $\gamma_t N_s$ AL frames, containing LT-coded output symbols, are generated from N_s video slices, where $\gamma_t \geq 1$ denotes the LT coding overhead at the *transmitter*. Note that the size of each LT-coded AL frame is still 150 bytes, i.e., the same as input video slice size, whereas the number of AL frames increases to $\gamma_t N_s$ from N_s. We emphasize that the transmitted LT overhead γ_t should not be confused with the received LT coding overhead γ_r. Generally, $\gamma_r \neq \gamma_t$ since some AL frames may not be correctly delivered to the receiver due to channel-induced losses.

The parameters of the UEP LT code at the AL are $k_i, i \in \{1, \ldots, 4\}$ and γ_t, which need to be optimized while considering the FEC at the PL in the cross-layer setup. Since all $r = 4$ priority levels have equal size, we have $\tau_1 = \tau_2 = \tau_3 = \tau_4 = \frac{1}{4}$ (see Section 3.1). For EEP/UEP LT coding, we use the standard degree distribution given by (1) [12,13,16].

When UEP rateless codes designed in [12,13] are used at the AL, all $\gamma_t N_s$ LT-coded symbols have equal importance. In other words, while more emphasis is given on the higher priority video slices, compared with the lower priority slices, in generating each encoded symbol, the UEP property is embedded in all the encoded symbols equally. Therefore, when UEP rateless codes designed in [12,13] are used, only EEP FEC coding should be performed at the PL. On the other hand, when video slices are passed to the lower layers without the AL FEC coding, the UEP FEC coding can be performed at the PL based on the slice priority. However, the rateless codes discussed in [21,25] are capable of encoded symbols with unequal importance.

4.3 Design of RCPC codes at the PL

At the PL, *cyclic redundancy check* (CRC) bits are added to each AL frame to detect any RCPC decoding errors. We use the industry-standard CRC-8 defined by the polynomial $1 + x^2 + x^4 + x^6 + x^7 + x^8$ [38]. Next, each AL frame is encoded using a UEP/EEP RCPC code. As mentioned earlier, we employ an RCPC code designed in [14] with the mother code rate of $R = \frac{1}{3}$ and memory of $M = 6$. Based on the AL frame priority level, the RCPC codes may be punctured to get appropriate higher rates. For four priority groups of AL frames, we have $R_1 \leq R_2 \leq R_3 \leq R_4$ and $R_i \in \left\{\frac{8}{8}, \frac{8}{9}, \frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}, \frac{8}{22}, \frac{8}{24}\right\}$, where R_i represents the RCPC code rate of priority i AL frames. Therefore, the parameters that need to be tuned at the PL are R_1 through R_4. For EEP RCPC codes, we have $R_1 = R_2 = R_3 = R_4$. We refer to a frame encoded by the RCPC code as a PL frame.

For the sake of simplicity and without the loss of generality, we assume that each transmitted packet contains one PL frame. Note that the number of PL frames in a packet does not affect the optimal cross-layer setup of FEC codes in our scheme. We have used a conventional BPSK modulation and a simple AWGN channel. Our model can be easily extended to the more complex channel models by using an appropriate P_d in (4) from [14]. To obtain the packet error rates at the PL on the receiver side, we first employ (4) to obtain the bit error rate of the received bitstream. Next, we employ Monte Carlo method to obtain the packet error rate at the receiver. We perform numerical RCPC encoding and CRC calculations and simulate the transmission. Finally, we find the ratio of correctly received packets by taking average over 10^3 packet transmissions in 10^3 iterations.

4.4 System model at transmitter

Based on our discussions so far, we can use four combinations of cross-layer FEC coding schemes at the AL and the PL (summarized in Table 2). Note that the FEC coding is necessary at the PL but optional at the AL. We illustrate the layout of cross-layer FEC schemes in Figure 1

Table 2 Various combinations of cross-layer FEC coding schemes

Model	S-I	S-II	S-III	S-IV
AL FEC	No FEC	No FEC	EEP	UEP
PL FEC	EEP	UEP	EEP	EEP

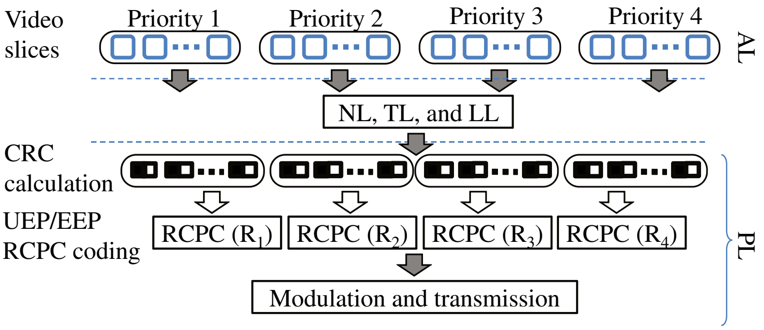

Figure 1 The proposed S-I and S-II cross-layer FEC schemes. In these schemes, the video slices are prioritized at the AL and UEP/EEP FEC coding is performed only at the PL. In S-I, we have $R_1 = R_2 = R_3 = R_4$. Here, TL, NL, and LL represent the transport, network, and link layers, respectively.

for S-I and S-II schemes and in Figure 2 for S-III and S-IV schemes. The cross-layer optimization of these FEC schemes is discussed in Section 5.

In S-I and S-II, FEC coding is applied only at the PL. In S-I, the equal protection (i.e., EEP RCPC coding) is provided to all frames regardless of their importance. In S-II, the video slices are protected at the PL with various protection levels based on their priority by using the UEP RCPC coding. We expect this scheme to have a considerably improved performance compared to S-I. Note that the priority of each AL frame is conveyed to the PL by using the cross-layer communication. This setup represents the schemes proposed in [36,39-45].

In S-III and S-IV, FEC coding is applied at both the AL and the PL in a cross-layer fashion. In S-III scheme, we add the FEC coding at the AL by using regular EEP LT codes to the base S-I setup. As we will see later, S-III cannot outperform S-I for all channel conditions since LT codes require extra coding overhead. However, this scheme has the ratelessness property, meaning that it can tolerate loss of the AL frames and still recover the original video slices after LT decoding. This is in contrast to S-I and S-II where the corrupted frames are considered lost. This setup represents the cross-layer FEC schemes proposed in [5,10,11,26-31,46].

In the proposed S-IV scheme, we apply the UEP LT codes where different slices are protected according to their priority. This scheme benefits both from ratelessness and UEP property. We expect this scheme to achieve the best performance. When LT coding is applied at the AL, the rateless coded symbols are uniformly generated and all the encoded AL frames have equal importance. As a result, using UEP FEC coding at the PL would not be beneficial. This is why we have used EEP FEC coding at the PL in the cross-layer S-III and S-IV schemes.

4.5 Decoding at receiver

Let PER_i denote the packet error rate of AL frames of priority i at the receiver after RCPC decoding and before LT decoding at the AL. PER_i can be computed using (3).

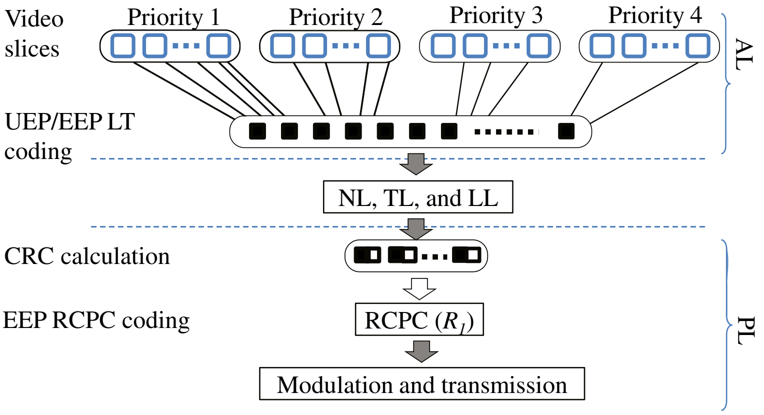

Figure 2 The proposed S-III and S-IV cross-layer FEC schemes. In these schemes, the video slices are prioritized at the AL and two layers of FEC coding at the AL and the PL are performed. We perform UEP/EEP LT coding at the AL and EEP RCPC coding at the PL. In S-III, we have $k_1 = k_2 = k_3 = k_4 = 1$ for EEP LT coding.

In S-I and S-II schemes, each AL frame consists of an uncoded video slice (i.e., LT coding is not performed at the AL). Therefore, the *video slice loss rate* (VSLR) of slices in priority i is $\mathrm{VSLR}_i = \mathrm{PER}_i$. In S-III and S-IV schemes, on the other hand, the LT decoding should also be performed, and the decoding error rate of LT codes should be considered in VSLR_i. In S-III and S-IV schemes, the EEP RCPC code is used at the PL; hence, we have PER_1=PER_2=PER_3=PER_4=PER. In this case, we employ (2) with $\gamma_r = \gamma_t N_s(1 - \mathrm{PER})$, degree distribution (1), and a given set of $k_i, i \in \{1, \ldots, 4\}$ to find the final LT decoding symbol error rates $y_i, i \in \{1, \ldots, 4\}$ for each priority at the receiver (see Section 3.1). If the symbol decoding error rate of priority i is y_i, then $\mathrm{VSLR}_i = y_i$.

5 Cross-layer optimization of the proposed FEC schemes

In our cross-layer FEC schemes, we consider the following issues. *First*, the AL and PL FEC codes share the same available channel bandwidth to add their coding redundancy. As the channel $\frac{E_s}{N_0}$ increases, the RCPC code rate at the PL can be increased. Thus, more channel bandwidth becomes available for improving the LT coding at the AL. For low values of $\frac{E_s}{N_0}$, assigning a higher portion of the available redundancy to LT codes at the AL may not improve the delivered video quality since almost all PL frames would be corrupted during transmission. Therefore, a stronger RCPC code rate should be used at the PL. This consumes a larger portion of the channel bandwidth allowing only a weaker LT code at the AL. *Second*, UEP FEC may be used either at the AL or the PL. We study how using UEP relates to varying $\frac{E_s}{N_0}$ and the bandwidth portions assigned to each FEC code. *Third*, the optimal FEC code rates for one scheme in Table 2 may be substantially different from another scheme.

To find the optimal parameters for both the FEC schemes and the portion of channel bandwidth they share, we discuss below the cross-layer optimization for the four schemes given in Table 2.

5.1 Formulation of optimization problem

The goal of cross-layer optimization in our scheme is to deliver a video with the highest possible PSNR for a given channel bandwidth C and SNR. Since computing the video PSNR requires decoding the video at the receiver, it is not feasible to use PSNR directly as the optimization metric due to its heavy computational complexity. The PSNR of a compressed video stream depends on several factors, including the video characteristics, bit rate, the percentage of lost slices, and their CMSE values [9,34]. Therefore, we define a function 'normalized F,' denoted by $\overline{F}$, which represents the weighted distortion contributed by the slice loss rates and their corresponding normalized CMSE values, as

$$\overline{F} = \sum_{i=1}^{r} \overline{\mathrm{CMSE}}_i^{\alpha} \mathrm{VSLR}_i. \tag{5}$$

Here, we use a parameter $\alpha \geq 0$ that needs to be *tuned* so that $\overline{F}$ can correctly capture the behavior of PSNR. For a compressed video whose PSNR for error-free transmission is already known, minimizing F results in minimizing the decrease in its PSNR. Selecting the optimal α is discussed in the next section.

To minimize $\overline{F}$, we tune the parameters of the FEC codes at the AL and the PL. In the S-I scheme, the optimization function finds the optimal RCPC code rate R for a given channel data rate C as

$$\begin{aligned} &\underset{\{R\}}{\operatorname{argmin}}\, \overline{F} = \{R^*\} \\ &\quad \text{s.t. } N_s(S+1)R^{-1} \leq C, \end{aligned} \tag{6}$$

where $S + 1$ is the slice size $S = 150$ bytes plus 1 byte of CRC.

In S-II, the optimization parameters are R_1 through R_4, such that $R_1 \leq R_2 \leq R_3 \leq R_4$. For this scheme, the optimization function can be written as

$$\begin{aligned} &\underset{\{R_1,R_2,R_3,R_4\}}{\operatorname{argmin}} \quad \overline{F} = \{R_1^*, R_2^*, R_3^*, R_4^*\} \\ &\quad \text{s.t. } N_s(S+1) \sum_{i=1}^{4} \frac{R_i^{-1}}{4} \leq C. \end{aligned} \tag{7}$$

The optimization parameters for S-III are γ_t and R. In S-III, we have $k_1 = k_2 = k_3 = k_4 = 1$ since EEP LT coding is used at the AL. The channel data rate is shared among the two FEC codes and needs to be tuned by selecting an appropriate γ_t. The optimization function is

$$\begin{aligned} &\underset{\{\gamma_t,R\}}{\operatorname{argmin}}\, \overline{F} = \{\gamma_t^*, R^*\} \\ &\quad \text{s.t. } \gamma_t N_s(S+1)R^{-1} \leq C. \end{aligned} \tag{8}$$

In S-IV, the UEP LT codes are used and optimization parameters are k_1 through k_3, along with γ_t and R. Here, the value of k_4 can be determined based on k_1 through

Table 3 PSNR of Bus video sequence for various values of α and $\frac{E_S}{N_0}$ with optimized *F* for S-II

$\frac{E_S}{N_0}$	**1 dB**		**2 dB**		**3 dB**		**4 dB**	
α	1,2	3	1,2	3	1	2,3	1	2,3
PSNR	18.2	16.85	22.3	19.8	25.8	20.6	29.69	29

The maximum achievable PSNRs are shown in italics.

Table 4 Optimal cross-layer parameters for S-I scheme with $C = 1.4$ Mbps

E_s/N_0	1 dB	1.25 dB	1.5 dB	1.75 dB	2 dB	2.25 dB	2.5 dB	2.75 dB	3 dB	4 dB	5 dB
$\overline{F}$	0.998	0.988	0.949	0.852	0.694	0.503	0.328	0.197	0.11	0.008	0
F_{Bus}	443.4	438.9	421.6	378.5	307.9	223.5	145.7	87.5	48.9	3.1	0
F_{Foreman}	214.7	212.5	204.1	183.3	149.1	108.2	70.6	42.4	23.7	1.5	0
$F_{\text{Coastguard}}$	179.8	178.0	171.0	153.5	124.9	90.6	59.1	35.5	19.8	1.3	0
R	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$
$\text{VSLR}_i, \forall i$	0.998	0.988	0.949	0.852	0.693	0.503	0.328	0.197	0.11	0.007	0

k_3 since $\sum_{j=1}^{r} k_j\tau_j = 1$ (see Section 3.1). As a result, the optimization function is

$$\underset{\{k_1,k_2,k_3,\gamma_t,R\}}{\text{argmin}} \quad \overline{F} = \{k_1^*, k_2^*, k_3^*, \gamma_t^*, R^*\} \qquad (9)$$
$$\text{s.t. } \gamma_t N_s (S+1) R^{-1} \leq C.$$

The optimization of the LT code's parameters involves employing (2) for various priority levels. Since (2) has a recursive form, it may not be represented by a linear function. Furthermore, the concatenation of two FEC codes presents a non-linear optimization problem, which cannot be solved using *linear programming* techniques. Therefore, we use the *genetic algorithms* (GA) to perform optimizations [47,48]. Although GA are computationally complex, they can give solutions which are close to the global optimum [47-49]. There are numerous implementations of GA. We used the GA toolbox available in Matlab [50]. We have provided a brief review on GA in the Appendix.

5.2 Optimal value of α

In Table 1, the normalized CMSE values ($\overline{\text{CMSE}}_i$) of the video sequences, except Akiyo, were similar. Therefore, the optimal parameters computed for the Bus video would be *almost* optimal for the other four video sequences generated by the same encoding parameters. We therefore use the $\overline{\text{CMSE}}_i$ of the Bus video with data rate of 840 kbps to perform our optimizations, followed by the Akiyo sequence. We implement our cross-layer FEC setup including LT coding at the AL and RCPC coding at the PL for S-I through S-IV (see Table 2) in Matlab environment.

We find the optimal value of α such that minimizing $\overline{F}$ maximizes the PSNR of the decoded video. For this, we perform the optimization to minimize $\overline{F}$ for various values of α and also compute the corresponding video PSNR. Note that the value of α has no effect on a cross-layer scheme with EEP FEC code since all VSLR_i's are equal in this case. Therefore, we perform our optimization for S-II, which is the simplest UEP FEC scheme. Table 3 reports the PSNR of the Bus video for three values of α and $\frac{E_S}{N_0}$ for $C = 1.4$ Mbps when $\overline{F}$ is minimized in S-II. The value of α that concurrently maximizes the PSNR of the video for all values of $\frac{E_S}{N_0}$ is $\alpha = 1$. Although not shown in Table 3, the non-integer values of α and $\alpha < 1$ were also considered in optimization. $\alpha = 1$ also gave the best results for Akiyo.

Table 5 Optimal cross-layer parameters for S-II scheme with $C = 1.4$ Mbps

E_s/N_0	1 dB	1.25 dB	1.5 dB	1.75 dB	2 dB	2.25 dB	2.5 dB	2.75 dB	3 dB	4 dB	5 dB
$\overline{F}$	0.172	0.163	0.158	0.111	0.077	0.059	0.05	0.046	0.041	0.003	0
F_{Bus}	76.1	72.2	70.1	49.3	34.0	25.9	22.1	20.4	17.9	1.1	0
F_{Foreman}	30.2	28.4	27.4	21.8	14.3	10.3	8.4	7.6	7.7	0.5	0
$F_{\text{Coastguard}}$	30.7	29.1	28.2	20.5	14.3	11.1	9.5	8.8	7.4	0.5	0
R_1	$\frac{8}{18}$	$\frac{8}{18}$	$\frac{8}{18}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$
R_2	$\frac{8}{16}$	$\frac{8}{16}$	$\frac{8}{16}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$
R_3	$\frac{8}{9}$	$\frac{8}{9}$	$\frac{8}{9}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$
R_4	1	1	1	1	1	1	1	1	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$
VSLR_1	0.007	0.003	0.001	0.072	0.036	0.017	0.008	0.004	0.001	0	0
VSLR_2	0.063	0.033	0.0162	0.072	0.036	0.017	0.008	0.004	0.11	0.007	0
VSLR_3	1	1	1	0.072	0.036	0.017	0.008	0.004	0.11	0.007	0
VSLR_4	1	1	1	1	1	1	1	1	0.11	0.007	0

Table 6 Optimal cross-layer parameters for S-III scheme with $C = 1.4$ Mbps

E_s/N_0	1.75 dB	2 dB	2.25 dB	2.5 dB	2.75 dB	3 dB	4 dB	5 dB
$\overline{F}$	1	0.972	0.268	0.022	0.021	0.017	0.007	0.006
F_{Bus}	444.3	431.9	119.2	9.8	9.3	5.3	2.1	0.8
F_{Foreman}	215.1	209.1	57.7	4.7	4.5	2.6	1.0	0.4
$F_{\text{Coastguard}}$	180.2	175.2	48.3	4.0	3.8	2.1	0.8	0.3
R	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{10}$	$\frac{8}{10}$	$\frac{8}{9}$
γ_t	1.05	1.05	1.05	1.05	1.05	1.25	1.25	1.4
$\text{VSLR}_i, \forall i$	1	0.972	0.268	0.022	0.021	0.012	0.005	0.002

5.3 Discussion of cross-layer optimization results

We report the cross-layer optimization results, including the FEC parameters (e.g., R_i, γ_t, and k_i), $VSLR_i$, normalized $\overline{F}$, and non-normalized F for the $\overline{\text{CMSE}}_i$ values of the Bus video. Note that F is calculated by replacing the $\overline{\text{CMSE}}_i$ by the actual average CMSE_i for the video sequence under consideration. The results of all four FEC schemes for three video sequences (Bus, Foreman, and Coastguard) are reported in Tables 4, 5, 6, and 7 for channel bit rate $C = 1.4$ Mbps. The results for Akiyo are discussed in Section 6.

From Tables 4 and 5, we observe that the use of UEP RCPC coding at the PL in the S-II scheme achieves much better performance (i.e., lower F_{Bus}) than the use of EEP RCPC coding in the S-I scheme. Both schemes do not use FEC coding at the AL.

Since the RCPC code rate of $\frac{8}{12}$ at the PL is not strong enough for $\frac{Es}{No} \leq 2$ dB, the value of F_{Bus} in the S-I scheme is high ($F_{\text{Bus}} > 300$ in Table 4) because many packet are corrupted due to high channel errors. For a successful decoding in LT, the number of error-free packets received should be above a threshold. As a result, the S-III scheme (which also uses RCPC with the same code rate as in S-I) achieves a lower performance (higher value of F_{Bus}) than S-I for $\frac{Es}{No} \leq 2$ dB (see Tables 4 and 6). However, the S-III scheme achieves much better performance ($F_{\text{Bus}} < 10$) than S-I for $\frac{Es}{No} \geq 2.5$ dB because fewer packets are now corrupted at the PL and the LT coding becomes effective.

From Tables 6 and 7, we observe that the proposed S-IV scheme achieves much lower values of F_{Bus} than S-III at all values of $\frac{Es}{No}$. This demonstrates that using UEP LT codes at the AL along with EEP RCPC codes at the PL gives a far superior performance than using EEP codes at both layers.

From Table 7 for the S-IV scheme, we observe an interesting trade-off between the code rates assigned to FEC codes at the AL and the PL. For lower values of $\frac{Es}{No}$, a larger portion of the bit budget is assigned to RCPC codes at the PL rather than LT codes at the AL because the LT coding cannot be effective when a large number of packets are corrupted due to channel errors. Furthermore, a stronger UEP (i.e., higher value of k_i to higher priority video slices) is provided at the AL. For higher values of $\frac{Es}{No}$, the RCPC

Table 7 Optimal cross-layer parameters for S-IV scheme with $C = 1.4$ Mbps

E_s/N_0	1 dB	1.25 dB	1.5 dB	1.75 dB	2 dB	2.25 dB	2.5 dB	2.75 dB	3 dB	4 dB	5 dB
$\overline{F}$	0.157	0.058	0.047	0.045	0.044	0.026	0.017	0.016	0.013	0.005	0.004
F_{Bus}	69.7	25.6	20.9	19.9	19.6	11.4	7.6	7.2	5.8	2.1	2.0
F_{Foreman}	27.3	10.1	7.8	7.3	7.2	5.1	3.4	3.2	2.6	0.9	0.9
$F_{\text{Coastguard}}$	28.0	10.9	9.0	8.6	8.5	4.7	3.1	2.9	2.4	0.9	0.8
R	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{10}$	$\frac{8}{10}$	$\frac{8}{10}$
γ_t	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.2	1.2	1.2
k_1	2	1.4	1.4	1.4	1.4	1.2	1.2	1.2	1.2	1.2	1.2
k_2	2	1.3	1.3	1.3	1.3	1.1	1	1	1	1	1
k_3	0	1.3	1.3	1.3	1.3	0.9	0.9	0.9	0.9	1	1
k_4	0	0	0	0	0	0.8	0.9	0.9	0.9	0.8	0.8
VSLR_1	0.004	0.014	0.004	0.002	0.002	0.015	0.008	0.008	0.006	0.002	0.002
VSLR_2	0.004	0.021	0.007	0.004	0.003	0.024	0.025	0.024	0.019	0.007	0.007
VSLR_3	1	0.021	0.007	0.004	0.003	0.064	0.043	0.041	0.034	0.007	0.007
VSLR_4	1	1	1	1	1	0.107	0.043	0.041	0.034	0.028	0.026

code rate is relatively high and more protection is provided to LT codes at the AL. Also, the UEP (i.e., value of k_i) at the AL is relatively less strong now.

Overall, the proposed S-IV scheme achieves the best performance at different channel SNRs, followed by the S-II scheme for $\frac{Es}{No} \leq 2.5$ dB. S-III outperforms S-II for other higher channel SNRs. We observe similar results for Foreman and Coastguard videos. Therefore, we can generally conclude that it is optimal to provide UEP at the AL and EEP at the PL using a cross-layer design.

Note that the optimization is performed only once for a given set of $\overline{\text{CMSE}}_i$ values, a GOP structure, and a set of channel SNRs, and need not be run separately for each GOP. The same set of optimized parameters can be used for any video stream with similar properties. Further, we should note that similar performance improvement is also observed for the 1.8-Mbps channel bit rate.

6 Performance evaluation of FEC schemes for test videos

In this section, we evaluate the performance of our optimized cross-layer FEC schemes for four CIF (352 × 288 pixels) video sequences: Bus, Foreman, Coastguard, and Akiyo. These sequences were encoded using H.264/AVC JM 14.2 reference software [51] at 840 kbps and 150 bytes slice size, for a GOP length of 30 frames with GOP structure *IDR B P B*...*P B* at 30 frames/s. The slices were formed using dispersed-mode FMO with two slice groups per frame. Two reference frames were used for predicting the *P* and *B* frames, with error concealment enabled using temporal concealment and spatial interpolation. We have used a channel transmission rate of $C = 1.4$ to study the performance over AWGN channels.

We used the slice loss rates reported in Tables 4 through 7 to evaluate the average PSNR of three video sequences (Bus, Foreman, and Coastguard) in Figures 3, 4, and 5. Figures 3, 4, and 5 confirm that our proposed cross-layer S-IV scheme, with UEP FEC coding at the AL and EEP FEC coding at the PL, achieves considerable improvement in average video PSNR, especially at low values of $\frac{E_s}{N_0}$. It outperforms the S-II scheme, which uses UEP RCPC code at the PL, by about 2 ∼ 7 dB for $\frac{E_s}{N_0} \leq 3.5$ dB. Only S-III has a comparable performance at $\frac{E_s}{N_0} \geq 2.5$ dB. However, at low values of $\frac{E_s}{N_0}$, the S-IV scheme considerably outperforms S-III.

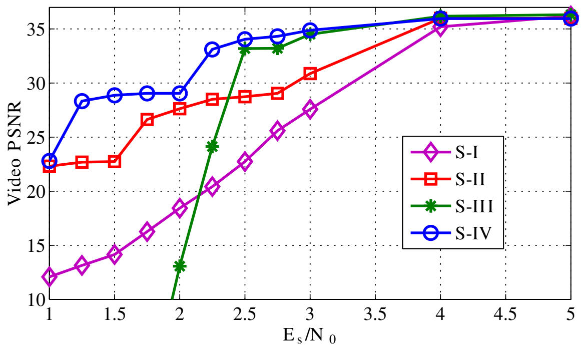

Figure 4 Average PSNR of Foreman video for different channel SNRs at $C = 1.4$ Mbps. The PSNR for the error-free channel is 36.9 dB.

Although our cross-layer FEC parameters were optimized for Bus sequences, the average PSNR performance is similar to that of the other two test video sequences, i.e., Foreman and Coastguard. As mentioned earlier, both sequences have different characteristics compared to the Bus sequence.

Since Akiyo has considerably different values of $\overline{\text{CMSE}}_i$, the proposed S-IV scheme designed by using Bus video's $\overline{\text{CMSE}}_i$ values would be suboptimal for Akiyo. In order to study the effect of these CMSE variations, we also designed the S-IV scheme by using the $\overline{\text{CMSE}}_i$ values of Akiyo and compare its performance with its suboptimal version. The optimization results are reported in Table 8. In this table, we also included the suboptimal values of

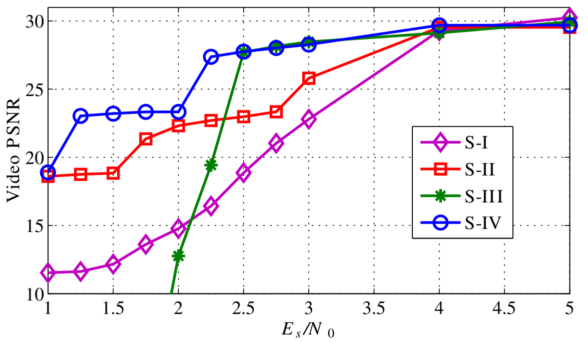

Figure 3 Average PSNR of Bus video for different channel SNRs at $C = 1.4$ Mbps. The PSNR for the error-free channel is 30.26 dB.

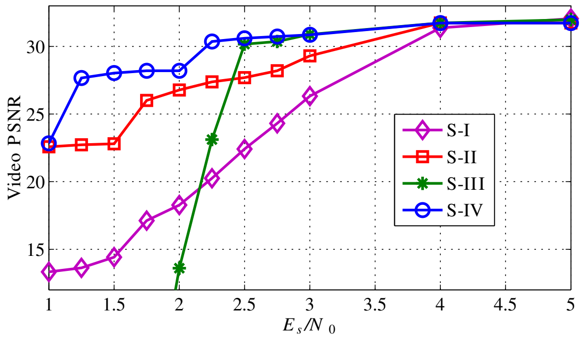

Figure 5 Average PSNR of Coastguard video for different channel SNRs at $C = 1.4$ Mbps. The PSNR for the error-free channel is 32.1 dB.

Table 8 Optimal cross-layer parameters for S-IV at $C = 1.4$ Mbps for Akiyo video sequence

E_s/N_0	1 dB	1.25 dB	1.5 dB	1.75 dB	2 dB	2.25 dB	2.5 dB	2.75 dB	3 dB	4 dB	5 dB
F_{opt}	1.111	0.600	0.287	0.243	0.229	0.223	0.221	0.219	0.215	0.066	0.062
F_{sub}	1.141	0.600	0.317	0.259	0.239	0.494	0.325	0.306	0.240	0.079	0.074
PSNR_{opt}	29.78	38.36	40.39	40.6	41.0	41.12	41.15	41.15	41.23	45.62	45.96
PSNR_{sub}	29.62	38.20	40.2	40.3	40.8	39.42	41.04	41.05	41.15	45.49	45.85
R	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{12}$	$\frac{8}{10}$	$\frac{8}{10}$	$\frac{8}{10}$
γ_t	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.2	1.2	1.2
k_1	2.3	1.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.3	1.3
k_2	1.7	1.3	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1	1
k_3	0	1.3	1	1	1	1	1	1	1	0.9	0.9
k_4	0	0	0	0	0	0	0	0	0	0.8	0.8
VSLR_1	0.001	0.014	0.001	0	0	0	0	0	0	0.001	0.001
VSLR_2	0.012	0.021	0.014	0.008	0.006	0.005	0.005	0.004	0.004	0.008	0.007
VSLR_3	1	0.021	0.039	0.024	0.018	0.016	0.015	0.015	0.013	0.015	0.014
VSLR_4	1	1	1	1	1	1	1	1	1	0.028	0.027

F_{sub} and PSNR_{sub}, which were obtained by using the optimized parameters of the Bus video from Table 7. The values of PSNR_{opt} and PSNR_{sub} are also shown in Figure 6.

In Table 8 (for optimal scheme) and Table 7 (for suboptimal scheme), the LT code overhead (i.e., γ_t) and RCPC code strength (R) are the same for both schemes, whereas the values of LT code protection level k_i for each priority class vary slightly (e.g., k_1 is higher for the optimal scheme compared to the suboptimal scheme). Similarly, the values of VSLR_i for higher priority slices (which have the most impact on F and PSNR) are similar in both tables, except for channel SNRs of 2.25, 2.5, and 2.75 dB in the decreasing order of the difference in values. The maximum PSNR degradation of the suboptimal scheme compared to the optimal scheme is 1.7 dB at the channel SNR of 2.25, with only about 0.1 to 0.3 dB PSNR degradation at other channel SNRs. We can, therefore, conclude that the performance of the proposed cross-layer FEC scheme is not very sensitive to the precise values of normalized CMSE.

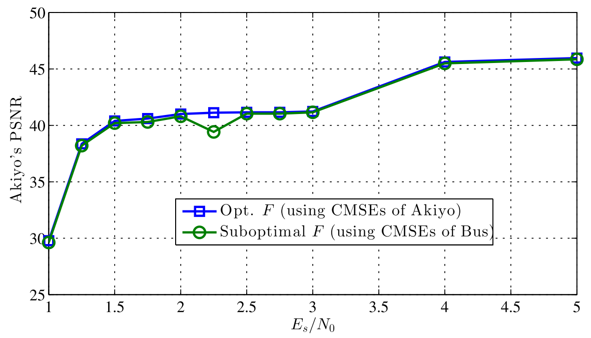

Figure 6 Average PSNR performance of the optimal and suboptimal cross-layer FEC schemes for Akiyo video sequence.

7 Conclusion

Previously, EEP and UEP FEC coding schemes have been used for video transmission over lossy channels. However, the joint optimization of cross-layer UEP FEC codes at the AL and the PL for robust video transmission has never been considered. In this paper, we used UEP LT coding at the AL and RCPC coding at the PL for robust H.264 video transmission over wireless channels. H.264 video slices were prioritized based on their contribution to video quality. We performed cross-layer optimization to concurrently tune the FEC code parameters at both layers, to minimize the video distortion, and to maximize the PSNR. We observed that our cross-layer FEC scheme outperformed other FEC schemes that use either UEP coding at the PL alone or EEP FEC schemes at the AL as well as the PL. Further, we showed that our optimization works well for different H.264-encoded video sequences, which have widely different characteristics.

Appendix

Introduction to genetic algorithms

J. Holland in [47] showed how the evolutionary process can be applied to solve a wide variety of problems using a parallel technique that is now called the genetic algorithms [48]. Non-linear and complicated optimization problems which cannot be solved employing conventional optimization algorithms such as *linear programming* can be effectively solved using genetic algorithms. Let W and $\overline{w} = \{w_1, w_2, \ldots, w_k\}$ denote the decision space and k decision variables, respectively. Let $F(\overline{w})$ denote the objective function that we need to optimize (minimize/maximize). In conventional genetic algorithms, each w_i is translated to a binary format. The steps to find the optimum answer are as follows:

1. Generate a random initial population of size i each including k members $\overline{w_j}, j = \{1, 2, \ldots, k\}$.
2. Translate the generated population from real numbers to a binary format considering desired precision.
3. Concatenate the translated version of k decision variables together to generate i binary population members.
4. Evaluate i fitnesses $F(\overline{w_j}, j \in \{1, 2, \ldots, i\})$ of the current population.
5. Select two parents randomly, assigning higher probability of selection to the parents with a better fitness value.
6. Perform *crossover* and *mutation* [47] on the parents to generate two offsprings. For crossover, cut two parents from a random location and exchange second parts to generate offsprings. For mutation, with a small probability, flip a random bit in the offsprings' bit streams.
7. Go to step 5 until $i - 2$ offsprings are generated.
8. Keep two parents with the best fitness values and replace the rest $i - 2$ with the new offsprings.
9. If maximum iterations are not reached, go to 4; otherwise, translate the member of population with the best fitness value from a binary to real format and report it as the final answer.

The above algorithm is an overall view of conventional genetic algorithms. However, many variations have been proposed since genetic algorithms were first introduced. For instance, the translation from real to binary and vice versa is no more performed, and the algorithm and the crossover and mutation are all performed in real numbers. More detailed explanation of genetic algorithms is out of the scope of this paper. We refer the interested readers for performance evaluations of genetic algorithm methods to [48,52] a and the numerous available surveys.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grants ECCS-1056065 and CCF-0915994, and by the Air Force Research Laboratory under award FA8750-11-1-0048.

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Air Force Research Laboratory.

Author details

[1] Department of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA. [2] Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182, USA. [3] Air Force Research Laboratory, Rome, NY 13441, USA.

Received: 16 November 2012 Accepted: 9 July 2013
Published: 13 August 2013

References

1. T Wiegand, GJ Sullivan, G Bjntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
2. T Wiegand, G Sullivan, *ITU-T, JVT, T Rec: H.264/IEC 14496-10 AVC 2003 Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification* (International Telecommunication Union, Geneva, 2002)
3. S Kumar, L Xu, MK Mandal, S Panchanathan, Error resiliency schemes in H.264/AVC standard,. Elsevier J. Vis. Commun. Image Representation Spec. Issue Emerg. H.264/AVC Video Coding Stand. **17**(2), 183–185 (2006)
4. T Stockhammer, M Hannuksela, T Wiegand, H.264/AVC in wireless environments. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 657–673 (2003)
5. T Stockhammer, A Shokrollahi, M Watson, M Luby, T Gasiba, Application layer forward error correction for mobile multimedia broadcasting, in *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T and MEDIAFLO* (Taylor & Francis, Boca Raton, 2008), pp. 239–280
6. M van der Schaar, S Shankar, Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. IEEE Wireless Commun. **12**(4), 50–58 (2005)
7. E Setton, T Yoo, X Zhu, A Goldsmith, B Girod, Cross-layer design of ad hoc networks for real-time video streaming. IEEE Wireless Commun. **12**(4), 59–65 (2005)
8. A Talari, N Rahnavard, Unequal error protection rateless coding for efficient MPEG video transmission, in *IEEE Military Communications Conference* (IEEE, Piscataway, 2009), pp. 1–7
9. S Paluri, K Kambhatla, S Kumar, B Bailey, P Cosman, JD Matyjas, Predicting slice loss distortion in H.264/AVC video for low complexity data prioritization, in *IEEE Int. Conf. Image Processing Proceedings (ICIP 2012),* (Orlando, 30 Sept–3 Oct 2012)
10. D Gomez-Barquero, A Bria, Application layer FEC for improved mobile reception of DVB-H streaming services, in *IEEE 64th Vehicular Technology Conference, VTC-2006 Fall* (IEEE, Piscataway, 2006), pp. 1–5
11. T Courtade, R Wesel, A cross-layer perspective on rateless coding for wireless channels, in *IEEE International Conference on Communications, ICC* (IEEE, Piscataway, 2009), pp. 1–6
12. N Rahnavard, B Vellambi, F Fekri, Rateless codes with unequal error protection property. IEEE Trans. Inf. Theory. **53**(4), 1521–1532 (2007)
13. N Rahnavard, F Fekri, Generalization of rateless codes for unequal error protection and recovery time: asymptotic analysis, in *IEEE Int. Symp. Inf. Theory* (IEEE, Piscataway, 2006), pp. 523–527
14. J Hagenauer, Rate-compatible punctured convolutional codes (RCPC codes) and their applications. IEEE Trans. Commun. **36**(4), 389–400 (1988)
15. M Luby, LT codes, in *The 43rd Annual IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, 2002), pp. 271–280
16. A Shokrollahi, Raptor codes. IEEE Trans. Inf. Theory. **52**(6), 2551–2567 (2006)
17. S Ahmad, R Hamzaoui, M Al-Akaidi, Adaptive unicast video streaming with rateless codes and feedback. IEEE Trans. Circuits Syst. Video Technol. **20**(2), 275–285 (2010)
18. P Cataldi, M Grangetto, T Tillo, E Magli, G Olmo, Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection. IEEE Trans. Image Process. **19**(6), 1491–1503 (2010)
19. H Kushwaha, Y Xing, R Chandramouli, H Heffes, Reliable multimedia transmission over cognitive radio networks using fountain codes. Proc. IEEE. **96**, 155–165 (2008)
20. C Hellge, T Schierl, T Wiegand, Receiver driven layered multicast with layer-aware forward error correction, in *15th IEEE International Conference on Image Processing, ICIP* (IEEE, Piscataway, 2008), pp. 2304–2307
21. D Vukobratovic, V Stankovic, D Sejdinovic, L Stankovic, Z Xiong, Expanding Window Fountain codes for scalable video multicast, in *IEEE International*

Conference on Multimedia and Expo (IEEE, Piscataway, 2008), pp. 77–80
22. AS Tan, A Aksay, C Bilen, GB Akar, E Arikan, Rate-distortion optimized layered stereoscopic video streaming with raptor codes, in *Packet Video* (IEEE, Piscataway, 2007), pp. 98–104
23. H Jenkac, T Stockhammer, Asynchronous media streaming over wireless broadcast channels, in *IEEE International Conference on Multimedia and Expo* (IEEE, Piscataway, 2005), pp. 1318–1321
24. S Ahmad, R Hamzaoui, M Al-Akaidi, Robust live unicast video streaming with rateless codes, in *Packet Video* (IEEE, Piscataway, 2007), pp. 78–84
25. D Vukobratovic, V Stankovic, D Sejdinovic, L Stankovic, Z Xiong, Scalable video multicast using expanding window fountain codes. IEEE Trans. Multimedia. **11**(6), 1094–1104 (2009)
26. M Luby, M Watson, T Gasiba, T Stockhammer, Mobile data broadcasting over MBMS tradeoffs in forward error correction, in *Proceedings of the 5th International Conference on Mobile and Ubiquitous Multimedia* (ACM, New York, 2006), p. 10
27. T Stockhammer, G Liebl, On practical crosslayer aspects in 3GPP video services, in *Proceedings of the International Workshop on Workshop on Mobile Video* (ACM, New York, 2007), pp. 7–12
28. J Afzal, T Stockhammer, T Gasiba, W Xu, Video streaming over MBMS: a system design approach. J. Multimedia. **1**(5), 25–35 (2006)
29. A Alexiou, C Bouras, A Papazois, A study of forward error correction for mobile multicast. Int. J. Commun. Syst. **24**(5), 607–627 (2011)
30. D Munaretto, D Jurca, J Widmer, Broadcast video streaming in cellular networks: an adaptation framework for channel, video and AL-FEC rates allocation, in *2010 The 5th Annual ICST Wireless Internet Conference (WICON)* (IEEE, Piscataway, 2010), pp. 1–9
31. C Bouras, V Kokkinos, A Papazois, Application layer forward error correction for multicast streaming over LTE networks. Int. J. Commun. Syst (2012). http://onlinelibrary.wiley.com/doi/10.1002/dac.2321/abstract
32. R Gallager, Low-density parity-check codes. Inf. Theory IRE Trans. **8**, 21–28 (1962)
33. N Thomos, S Argyropoulos, N Boulgouris, M Strintzis, Robust transmission of H. 264/AVC streams using adaptive group slicing and unequal error protection. EURASIP J. Appl. Signal Process. **2006**, 120–120 (2006)
34. K Kambhatla, S Kumar, P Cosman, Wireless H.264 video quality enhancement through optimal prioritized packet fragmentation. IEEE Trans. Multimedia. **14**(5), 1480–1495 (2012)
35. S Kumar, A Janarthanan, MM Shakeel, S Maroo, JD Matyjas, M Medley, Robust H.264/AVC video coding with priority classification, adaptive NALU size and fragmentation, in *IEEE MILCOM Proceedings, 2009,* (Boston, 18)
36. E Baccaglini, T Tillo, G Olmo, Slice sorting for unequal loss protection of video streams. IEEE Signal Process. Lett. **15**, 581–584 (2008)
37. S Argyropoulos, A Tan, N Thomos, E Arikan, M Strintzis, Robust transmission of multi-view video streams using flexible macroblock ordering and systematic LT codes, in *3DTV Conference, 2007* (IEEE, Piscataway, 2007), pp. 1–4
38. P Koopman, T Chakravarty, Cyclic redundancy code (CRC) polynomial selection for embedded networks, in *International Conference on Dependable Systems and Networks* (IEEE, Piscataway, 2004), pp. 145–154
39. A Bouabdallah, J Lacan, Dependency-aware unequal erasure protection codes. J. Zhejiang Univ. Sci. A. **7**, 27–33 (2006)
40. E Maani, A Katsaggelos, Unequal error protection for robust streaming of scalable video over packet lossy networks. IEEE Trans. Circuits Syst. Video Technol. **20**(3), 407–416 (2010)
41. W Xiang, C Zhu, CK Siew, Y Xu, M Liu, Forward error correction-based 2-D layered multiple description coding for error-resilient H.264 SVC video transmission. IEEE Trans. Circuits Syst. Video Technol. **19**(12), 1730–1738 (2009)
42. H Ha, C Yim, Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC. IEEE Trans. Consum. Electron. **54**(2), 736–744 (2008)
43. Y Liu, S Yu, Adaptive unequal loss protection for scalable video streaming over IP networks. IEEE Trans. Consum. Electron. **51**(4), 1277–1282 (2005)
44. S Yingbo Shi, W Chengke Wu, D Jianchao Du, A novel unequal loss protection approach for scalable video streaming over wireless networks. IEEE Trans. Consum. Electron. **53**(2), 363–368 (2007)
45. XJ Zhang, XH Peng, R Haywood, T Porter, Robust video transmission over lossy network by exploiting H.264/AVC data partitioning, in *5th International Conference on Broadband Communications, Networks and Systems, BROADNETS* (IEEE, Piscataway, 2008), pp. 307–314
46. T Gasiba, W Xu, T Stockhammer, Enhanced system design for download and streaming services using Raptor codes. European Trans. Telecommun. **20**(2), 159–173 (2009)
47. J Holland, *Adaptation in Natural and Artificial Systems* (MIT Press, Cambridge, 1992)
48. JR Koza, Survey of genetic algorithms and genetic programming, in *WESCON/'95. Conference Record* (IEEE, Piscataway, 1995), p. 589
49. D Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers* (World Scientific, Singapore, 1999)
50. MathWorks, *Global Optimization Toolbox: User's Guide (R2011b)* (MathWorks, Natick, 2011)
51. JVT, H.264/AVC Reference Software JM14.2. ISO/IEC Std. http://iphome.hhi.de/suehring/tml/download/. Accessed 12 Feb 2012
52. K Deb, A Pratap, S Agarwal, T Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

doi:10.1186/1687-1499-2013-206
Cite this article as: Talari *et al.*: Optimized cross-layer forward error correction coding for H.264 AVC video transmission over wireless channels. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:206.