

Research Article

Application Layer Systematic Network Coding for Sliced H.264/AVC Video Streaming

Sajid Nazir,¹ Vladimir Stanković,¹ Ivan Andonović,¹ and Dejan Vukobratović²

¹Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XQ, UK

²Department of Power, Electronics and Communication Engineering, University of Novi Sad, 21000 Novi Sad, Serbia

Correspondence should be addressed to Sajid Nazir, nazirsajid@yahoo.com

Received 14 October 2011; Revised 22 March 2012; Accepted 5 April 2012

Academic Editor: Vaggelis Kapoulas

Copyright © 2012 Sajid Nazir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Application Layer Forward Error Correction (AL-FEC) with rateless codes can be applied to protect the video data over lossy channels. Expanding Window Random Linear Codes (EW RLCs) are a flexible unequal error protection fountain coding scheme which can provide prioritized data transmission. In this paper, we propose a system that exploits systematic EW RLC for H.264/Advanced Video Coding (AVC) slice-partitioned data. The system prioritizes slices based on their PSNR contribution to reconstruction as well as temporal significance. Simulation results demonstrate usefulness of using relative slice priority with systematic codes for multimedia broadcast applications.

1. Introduction

H.264 Advanced Video Coding (AVC) [1] is currently the most commonly used video coding standard, which is gaining widespread use in the emerging communication standards and applications.

Two key challenges of multimedia communication applications over wireless networks are high and varying error characteristics of underlying communications channels and huge heterogeneity of users' equipment.

One of the solutions is to use channel coding techniques which could recover the original data despite losses. The latest state-of-the-art solutions like those based on Reed Solomon (RS) codes are inflexible because the code rate has to be fixed in advance. Moreover, the encoding and decoding operations are quite complex especially for large Galois Field. For such codes, the error characteristics of the channel must be known in advance in order to adjust the code rate to it. This solution does not extend well to multiple receivers as then only a worst-case erasure channel can be assumed for all receivers.

To enable communications in the presence of packet losses, rateless Digital Fountain Raptor codes [2] have become standardized solution in many wireless systems

such as Digital Video Broadcasting-Handheld (DVB-H) [3–5], Multimedia Broadcast Multicast Service (MBMS), and mobile Worldwide Interoperability for Microwave Access (WiMax) [6].

Another class of rateless codes which have been gaining increased popularity for applications in wireless broadcast/cellular networks are Random Linear Codes (RLCs) [7, 8]. RLCs show near-capacity performance over erasure channels even for low codeword lengths [9, 10]. In addition, the emerging networking concepts, such as hybrid broadcast/cellular networks (with users equipped with multiple interfaces) or device-to-device communications, offer a number of opportunities for achieving network coding gains using RLC [11].

Traditional solutions for reliable multimedia delivery use multiple independent Reed Solomon (RS) codes with different rate allocation over importance classes [12, 13]. These solutions do not have the rateless property and thus have to be designed for the worst channel conditions, and they cannot explore network coding gains over network topologies by packet processing in intermediate nodes.

For applications where short message lengths represent a natural choice (such as multimedia delivery) and where wireless multihop communications are encountered, the RLC

scheme represents a more efficient and versatile approach. Such realizations are expected to result in increased throughput of wireless multihop broadcast/cellular networks [8]. Due to all this, RLCs have been considered as a unique rateless/network coding solution [14–18] for emerging wireless systems, such as Long-Term Evolution-Advanced (LTE-A) and DVB-NGH (Next Generation Handheld). (See [19] for performance/complexity comparison between Raptor codes [2] and RLCs.)

The inherent disadvantage of RLC is that RLC suffers from high decoding complexity of the Gaussian Elimination decoding as codeword length increases. However, even with very short codeword lengths that admit efficient implementation, RLC performance matches that of Raptor codes of higher codeword lengths [19, 20].

Both Raptor codes and RLC are “all or nothing” codes that equally protect the entire stream. For embedded and scalable sources where different parts of the stream have different importance to reconstruction, unequal error protection (UEP) is beneficial. Expanding Window Fountain (EWF) codes as a class of UEP FEC codes for scalable video delivery are proposed in [21]. EWF codes are based on the idea of creating a set of “nested windows” over a source block. This EW concept is extended to RLC, and performance limits are presented in [22]. The suitability of nonsystematic EW RLC for transmission of data-partitioned H.264/AVC has been investigated in [4].

As compared to the data partitioning H.264/AVC feature [1], slicing has an advantage that the size of slices can be tailored to the application. The slicing feature of H.264/AVC can be used to partition video stream into classes of decreasing importance (for video reconstruction) with a very small decrease in overall performance.

A scheme has been proposed in [23] based on macroblock classification into three slice groups and UEP of H.264/AVC streams. The ordering of macroblocks into three slice groups is done by examining their contribution to the video quality. The three slice groups are then protected with UEP using RS coding. In [24], a slice sorting by relevance (SSR) algorithm for prioritizing slices based on their contribution to the reconstruction is used together with RS coding. The work in [24] is later extended in [25] and proposes an algorithm termed Concealment Driven Slice Ordering with RS codes. The ordering of slices is based on error propagation effect and the rate devoted to each slice.

The proposed work differs from the earlier work in the method of prioritizing slices and choice of systematic rateless codes for channel coding. The slice-partitioned video stream can provide an advantage with respect to H.264 Scalable Video Coding (SVC) [26] of better coding efficiency and compliance with the AVC standard. The layered video can be protected by EW RLC codes that can provide a different degree of protection to each layer/window.

Building on our prior work [4, 19, 20, 22], the focus of this study is to analyse the use of the EW approach with systematic RLC as component codes for UEP of the slice-partitioned H.264/AVC video. Systematic RLCs have the advantage of supporting more efficient encoding and decoding procedures compared to nonsystematic RLC.

In contrast to [24], where priority layers are built based purely on distortion information, in this paper, we propose a new cost function that takes into account the frame play out deadline and temporal error propagation to better prioritize slices into quality layers.

Our simulation results show that EW RLC can be used to effectively protect the different priority windows for reliable video transmission over packet erasure channels. Significant performance gains are obtained compared to the equal error protection scheme and the benchmark scheme that prioritizes the sliced stream in an ad hoc fashion.

The rest of the paper is structured as follows. The relevant background on RLC and the slicing feature of H.264/AVC is covered in Section 2. The proposed system is described in Section 3. The results are presented in Section 4 and conclusion and future research in Section 5.

2. Background

In this section, we give a brief background on slicing in H.264/AVC [1, 27] and overview of RLC [7] and EW RLC [22] coding scheme.

2.1. Slicing in H.264/AVC. H.264/AVC provides many error-resilience features to mitigate the effect of lost packets during transmission.

One such scheme available in the baseline profile is slicing [27], which enables the partitioning of a frame into two or more independently coded sections, called slices. Each slice in a frame can have either a fixed number of assigned macroblocks (MBs) or fixed data rate. Each coded slice is independently decodable; however, the slices have different contribution (importance) to the video reconstruction. Thus, arranging the slices in decreasing order of their contribution to reconstruction can be used to provide a layered video stream suitable for UEP.

2.2. Random Linear Codes (RLCs). RLC applied over a source message produces encoded packets as random linear combinations of message packets with coefficients randomly selected from a given finite field $GF(2^q)$. For example, using RLC over a source message x of length K , an encoded packet ω is obtained as $\omega = \sum_{i=1}^K \alpha_i \cdot x_i$, where α_i is a randomly selected element of $GF(2^q)$. The resulting encoded packet ω is of the same length (b bits) as the source message packets. In addition, to each encoded packet, ω a header information is attached that contains the so-called global encoding vector $\mathbf{g} = \{\alpha_1, \alpha_2, \dots, \alpha_K\}$ consisting of randomly selected finite field coefficients. The header requirements in a unicast point-to-point setup can be relaxed if a pair of synchronized random number generators (RNGs) is used at the transmitter and the receiver and only the RNG seed is communicated within each encoded packet header. The encoding procedure is repeated at the transmitter in a rateless fashion.

Thus, each encoded symbol is a linear combination of all or a subset of the original source symbols. An RNG seed

carried in the header of the encoded symbol can be used at the decoder to recover the coefficients used at the encoder.

Encoding procedure is simple to implement, and, for sufficiently large finite field used for creating linear combinations of source symbols, RLC codes perform as near-optimal erasure codes (one-byte field GF(256) is usually sufficiently good [7]).

For practical network coding, RLC is used at source nodes for encoding the source message packets and at intermediate network nodes for random recombining of incoming and/or buffered encoded packets. The source nodes and intermediate nodes may produce encoded packets in a rateless fashion, until the requirements of receiving nodes are met (which may be confirmed by feedback messages), or in the delay-constrained applications, until a new source block is scheduled for transmission.

After sufficient linearly independent coded symbols have been received, the decoder can recover the original source symbols.

The RLC use is hindered by the decoding complexity of Gaussian Elimination decoding, which is polynomial in the number of symbols. However, for short lengths of the source messages, the decoding complexity is acceptable (see [19, 28] and references therein).

A systematic code is any error-correcting code in which the input data is embedded in the encoded symbol. The advantage of such codes is that the receiver does not need to recover the original source symbol in case of correct reception.

When erasure rates are low, it is effective to use systematic RLC, which further reduce the decoding complexity, since the received systemic packets can be used to reduce the effective code length before Gaussian Elimination decoding.

2.3. EW RLC [22]. In [21], EWF codes as a class of UEP fountain codes are proposed. EWF codes are based on the idea of creating a set of “nested windows” over the source block. The rateless encoding process is then adapted to use this windowing information while producing encoded packets. In this paper, we use the main concept of EWF to create EW RLC [22] from consecutive source blocks containing fixed number of symbols (data packets).

First, we define a set of windows over the groups of source symbols of unequal importance. Coding is then performed over progressively increasing source block subset windows aligned with this “most to least importance” subsets.

The general layout of a window structure with three importance classes is shown in Figure 1. The window with the most important subset of encoded data is W_1 , and the importance of data additionally included in windows progressively decreases as we proceed to W_3 . The subset data of W_1 is contained in all the subsequent windows and is hence the best protected. Apart from W_1 , each window in addition to some of its own data also encloses all the data of the higher importance windows. Conventional RLC is applied on each window.

The encoding process for EW RLC has one important initial step that is to first select a window from which

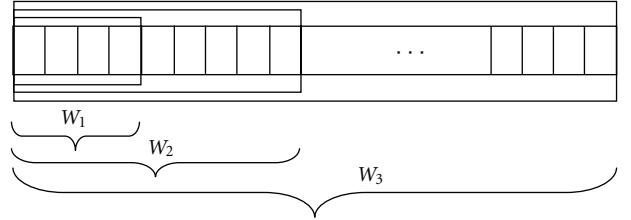


FIGURE 1: Expanding window structure with three windows.

the RLC encoded symbol is to be generated. This selection of a window is determined by probability of selection of a window which is a preassigned parameter keeping in mind the importance of different layers and the data rate available. After a window is selected, the encoding is the standard RLC encoding performed over the source packets contained in that particular window only [22]. The window selection procedure is independently repeated for each created encoded packet.

In [22], analytical performance of EW RLC is given together with comparison with traditional nonoverlapping UEP RLCs that use independent code for each window. In [4, 20], nonoverlapping window (NOW) and EW have been used to provide unequal error protection to the data partitioned H.264/AVC video data. It is shown in [4] that the performance of EW is better as compared to NOW because in NOW each window is independently decoded and thus the low priority windows do not contribute to recovery of the high priority windows.

3. The Proposed System

In this section, we propose a system for optimally protecting the slice-partitioned H.264/AVC video data with systematic EW RLC. We assume that the encoded video stream is transmitted over a packet loss channel. That is, all packets that arrive at the application-layer RLC decoder are correct, while those with bit errors are discarded by error detection codes, such as Cyclic Redundancy Check (CRC) codes present at the lower layer in the protocol stack (e.g., physical or link layer). We further assume that error detection capability of the employed CRC codes is perfect, which is usual assumption [7, 9–11, 24]. Thus, the application layer-to-application layer channel is modeled as packet erasure channel with random packet drop statistics.

In order to increase error resilience, we encode a video sequence using slicing with a fixed slice size of 600 bytes. That is, after the H.264/AVC encoding, we obtain the video data in which each frame including the IDR (instantaneous decoder refresh) is divided into slices of 600 ± 3 bytes, except for the last slice of each frame which can have a lesser size. The size of 600 bytes is chosen here to keep the number of RLC symbols per codeword low in order to reduce the decoding complexity of Gaussian Elimination. See [11, 28] for discussion about acceptable block lengths for real-time RLC decoding. The resulting slices carry different importance to reconstruction which has been used to achieve UEP (see [24, 29] and references therein).

After source coding, EW RLC coding takes place. Since systematic RLCs are used, first all encoded symbols (from all the slices) are transmitted without any coding. Because of possible errors/erasures in the channels, some packets will be missing at the decoder.

To correct these erasures, RLC redundancy packets are generated next.

Before RLC, the priority of each slice is obtained by dropping it from the Group of Pictures (GOPs) data and measuring the resulting peak signal-to-noise ratio (PSNR), as a frame-by-frame average of the entire GOP, by actual decoding. This also takes into account the error propagation effect to the subsequent frames due to loss of a slice in an earlier frame. That is, the cumulative PSNR of the GOP is measured by dropping each slice in turn starting at the first P frame. After having obtained the cumulative PSNR values for each slice (as dropped), the difference from the full-decoding PSNR of the GOP is measured. Determining PSNR drop can easily be done during the encoding process with negligible added complexity (see [24]).

The results are shown in Figure 2 for the first GOP (having 16 frames and the encoding structure IPPPP...) of the standard CIF Foreman sequence. It can be seen from Figure 2 that the importance of the slices on total frame-averaged PSNR generally decreases as we move towards the end of the GOP. Similar results are shown in Figure 3 for the first GOP (having 64 frames and the encoding structure IPPPP...) for the Paris sequence. As can be seen from the figures, the PSNR drop values for Paris sequence are larger due to large GOP size.

Thus, we can sort the slices into multiple priority layers and assign a higher degree of protection to the important layers as compared to the layers containing less significant slices. Such layering enables a prioritized data transmission with UEP schemes and was used before in [24, 29].

Purely grouping the slices into priority classes based on the PSNR decrease shown in Figure 2, as done in [24, 29], does not take into account real-time frame playout deadline (frames coming sooner should be given a higher priority).

Motivated by this, we redefine a cost function used in [29], to take into account not only the drop in cumulative PSNR for each slice, but also the temporal importance of a slice:

$$W(\text{slice}) = D(\text{slice}) - w * \tau(\text{slice}), \quad (1)$$

where $D(\text{slice})$ represents the drop in the cumulative PSNR (see y -axis of Figure 2). The value of $\tau(\text{slice})$ represents the playout time deadline of the slice (frame) relative to the playout time of the first IDR in the GOP. That is, $\tau(\text{slice})$ of the IDR frame is set to zero, and each subsequent frame adds its playout time duration to this value. w is a constant that trades off the distortion $D(\text{slice})$ and remaining playout time.

In this way, we create a system to assign a priority to all the slices trading off importance of the slices to reconstruction and playout time deadline. After computing $W(\text{slice})$ for all the slices in a GOP and selecting threshold values $T_1 > T_2 > \dots > T_{L-1}$, we group the slices into L layers.

The first layer includes IDR and slices with $W(\text{slice}) \geq T_1$, the second layer includes all remaining slices with $W(\text{slice}) \geq T_2$, and so forth. In addition, our algorithm also puts at least one slice per frame to the first layer, if none is selected (from a frame) based on the above criteria alone. This helps to stop the error propagation effect further and thus improves resulting PSNR. Such selections may be needed for frames which occur towards the end of GOP as can be seen from Figure 2.

In the proposed scheme, we can create L windows using a threshold $(L-1)$ -tuple T_1, \dots, T_{L-1} , and allocate different protection to each window. Note that the slices would already be in their decoding order within each layer. However, within each window, the slices will need to be restored to the original order to enable decoding by the AVC decoder.

After determining thresholds and assigning slices to the L windows, the size of each layer is fixed. Then, the remaining task is to find the optimal allocation of redundancy to each layer, or equivalently probability of window selection. We express the probability of window selection as an L -tuple where i th entry denotes probability of selection of a packet from layer i . For example, let $L = 2$ and the vector of selection probabilities as $[0.6, 0.4]$, this implies that the first window, W_1 , will have a selection probability of 0.6, whereas W_2 will have a selection probability of 0.4. That is, in average, redundant packets from W_1 will comprise 60% of the overall redundancy.

To find optimal packet selection vector, we maximize the expected PSNR using analytically computed probabilities of decoding error performance. That is,

$$\max_{\pi} \text{PSNR}(\pi) = \sum_{i=0}^L P(i) \text{psnr}(i), \quad (2)$$

where $P(i)$ is probability that layer i will be the highest layer recovered, $P(0)$ is the probability that nothing is recovered, $\text{psnr}(i)$ is the PSNR of the reconstruction if all layers up to and including layer i are recovered, π is an L -tuple vector of window selection probabilities that determines the UEP allocation scheme, and $\text{PSNR}(\pi)$ is the expected PSNR when UEP scheme π is used.

In the above maximization, we made assumption that, if decoding of window i fails, none of the packets from window $j \geq i$ can be used for reconstruction. This is true for nonsystematic EW RLC and approximation for systematic EW RLC.

Analytical expressions for probabilities $P(i)$ assuming a random channel loss model for EW RLC as derived in our prior work [22] are as follows:

$$P(l) = \begin{cases} 1 - P_{d,N}(1), & l = 0, \\ \prod_{i=1}^l P_{d,N}(i) \cdot (1 - P_{d,N}(l+1)), & 1 \leq l \leq L-1, \\ \prod_{i=1}^L P_{d,N}(i), & l = L, \end{cases} \quad (3)$$

where the desired decoding probabilities $P_{d,N}(l)$ are expressed as

$$P_{d,N}(l) = \sum_{\substack{(n_1, n_2, \dots, n_L): \\ 0 \leq N_1 \leq N_2 \leq \dots \leq N_L = N}} P_{r(\xi),N}(n) P_{d,N}(l | n). \quad (4)$$

The expression for $P_{r(\xi),N}$ is as given:

$$P_{r(\xi),N}(n) = \frac{N!}{n_1! n_2! \cdots n_L!} \Gamma_1^{n_1} \Gamma_2^{n_2} \cdots \Gamma_L^{n_L}, \quad (5)$$

whereas

$$\begin{aligned} & P_{d,N}(l | n) \\ &= P_r(R_l = K_l) + P_r(R_{l+1} = K_{l+1}, R_l < K_l) \\ &+ \cdots + P_r(R_L = K, R_l < K_l, R_{l+1} < K_{l+1} \cdots R_{L-1} < K_{L-1}), \end{aligned} \quad (6)$$

where Γ_i is the probability of selection of the i th window, R_i s are random variables denoting the number of received packets from window i , and K_i is the number of source packets in window i .

The optimization method is exhaustive search and scales linearly with the number of UEP schemes being used.

For error concealment, we repeat the last correctly decoded frame to replace frames for which the base layer is not decoded properly.

4. Results and Analysis

In this section, we present our simulation results. For simplicity, we consider the case of $L = 2$ layers: high-priority layer (HPL) that contains more important slices, whose $W(\text{slice}) \geq T$, and low-priority layer (LPL) that contains less important slices for which $W(\text{slice}) < T$, where T is the chosen threshold.

The thresholds determine the source rate for each layer. For example, a lower T would result in a lower source rate (and, hence, error-free performance) for the base layer. Thus, T_i s are set based on available clients' bandwidths as well as desired error-free performance levels. In practice, transmitter can dynamically adapt the source rate per layer to varying channel conditions of different clients by changing T_i 's.

The video sequence *Foreman* in the CIF format is encoded using the H.264/AVC software JM version 16.2 [30]. First, we use the GOP size of 16 frames with a frame structure IPPP..., with a fixed slice size of 600 bytes. We compare three schemes: one is the proposed UEP scheme optimized using (1). The second scheme is the benchmark scheme, where we put all the slices of IDR and the first slice of each frame in HPL and all other slices in LPL. The third is the equal error protection (EEP) scheme that protects all slices equally. Note that the benchmark scheme is a low-complexity scheme where prioritization is done in an ad hoc manner; it still uses the same systematic EW RLC for protection of the two layers.

The proposed scheme is designed in accordance with the algorithm described in Section 3 with $T = 0.78$ and $w = 2.5$. The sizes, number of packets (same as the number of slices in a layer), and resulting PSNR values for both configurations are shown in Table 1.

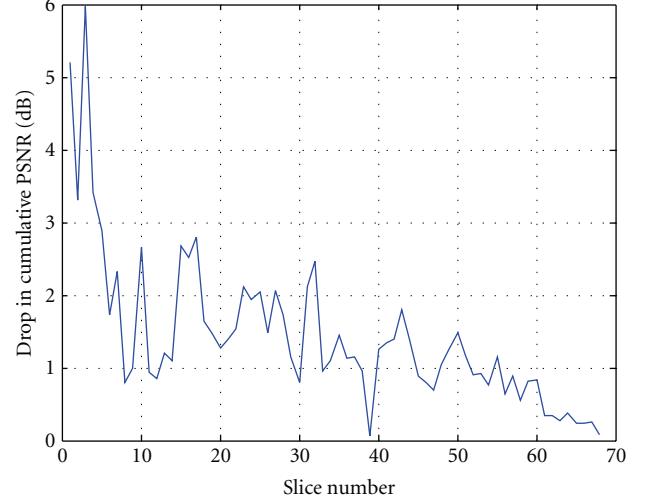


FIGURE 2: Drop in PSNR for non-IDR slices-*Foreman* sequence GOP16.

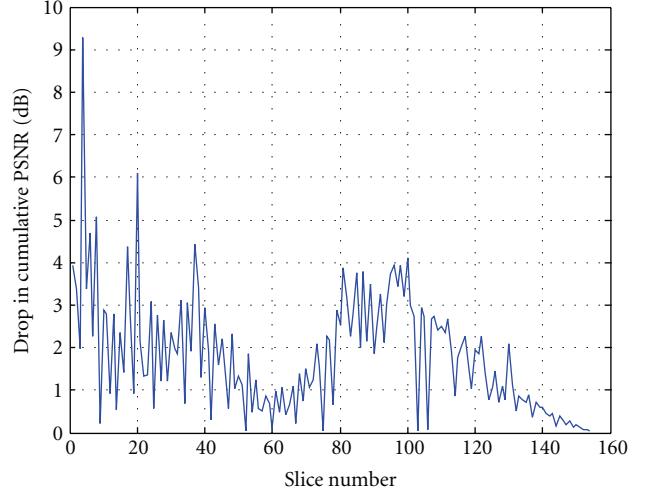


FIGURE 3: Drop in PSNR for non-IDR slices-*Paris* sequence GOP64.

For this selection of T , the proposed UEP scheme has larger HPL than the benchmark.

Note, however, that a smaller HPL for the proposed scheme could be obtained by suitably selecting parameter T and w in (1).

All schemes are compared at the same transmission bitrate. For an L -layer scheme, the overhead cost needed to describe a UEP solution is $7 \times L$ plus $(L - 1) * 8 * 2$ to convey T_i s and w_i s. For $L = 2$ used in this paper, this number is only 30 bits and has not been taken into consideration.

The proposed schemes are simulated with transmission of EW RLC for 1000 runs and the results averaged. The total number of packets to be transmitted for each run is 100. Because of the employed systematic RLC, the transmission takes place in two phases. In Phase I, we transmit 77 packets consisting of the source symbols. In Phase II, we transmit additional packets in accordance with EW RLC. Note that Phase I will be the same for all the three schemes, whereas,

TABLE 1: Layer sizes and PSNR contributions for $T = 0.78$ and $w = 2.5$ for Foreman sequence.

Layer	Proposed			Benchmark		
	Size (bytes)	Pkt	PSNR (dB)	Size (bytes)	Pkt	PSNR (dB)
HPL	21818	42	27.6	13042	24	23.14
LPL	19218	35	36.39	27994	53	36.39
Total	41036	77	36.39	41036	77	36.39

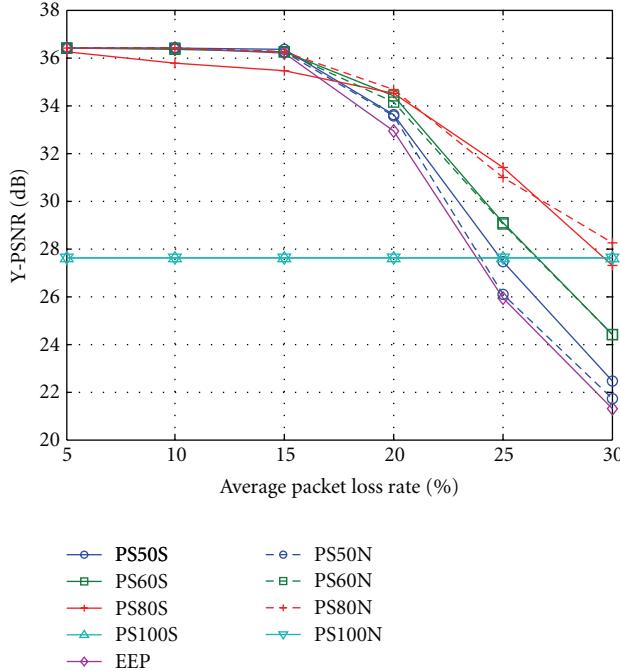


FIGURE 4: Comparison between systematic and nonsystematic EW RLC codes.

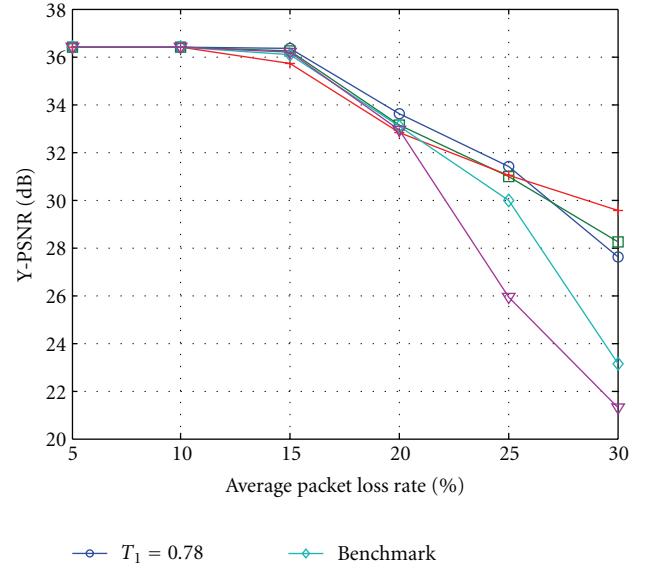
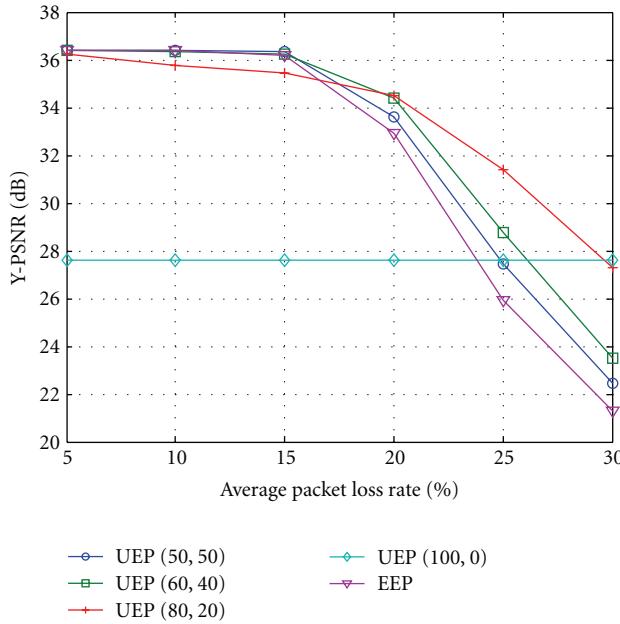
FIGURE 6: Optimized results for three values of T for the Foreman sequence.

FIGURE 5: PSNR versus average PLR for the proposed scheme and the EEP scheme.

Phase II, the probability of selection can govern a prioritized transmission of HPL. The important phenomenon seen here is that, since each slice is independently decodable, the PSNR obtained in the case when RLC decoding of LPL fails and decoding of HPL succeeds is higher than the PSNR of successfully decoded HPL due to useful packets that are received from LPL during Phase I.

This gain comes from the correct reception of additional LPL symbols from Phase I even with failure of LPL decoding. The simulations have been performed for different packet loss rates (PLRs) and different probabilities of window selection to evaluate the performance of the slicing feature to overcome losses.

In the case of nonsystematic codes, if the first window W_1 (or W_2) does not get decoded, the entire GOP is considered to be lost. However, in case of systematic codes, it is still possible for the H.264/AVC decoder to decode the GOP as long as its IDR frame has been received correctly. In case of loss of IDR with systematic codes, the entire GOP is lost. The PSNR for such cases is obtained by using the last frame of the previously decoded GOP to replace all frames of the lost GOP.

The various configurations are used to create different UEP schemes based on protecting the constituent windows

with different protection, based on probabilistically selecting a window for each output symbol at the transmitter. An increase in the selection probability of window 1 (W_1) will improve its robustness at a cost of a decrease in robustness of the succeeding layer(s). The EEP scheme is the case where only the largest window is selected with 100% probability. This means that all of the data is protected with no preference for the data considered important, that is, window W_1 .

In Figure 4, we present the results of comparison between the systematic codes and nonsystematic codes. The scheme PS60S is a scheme with probability of W_1 selection equal 0.6 (i.e., probability to select a symbol from HPL is 0.6), and the suffix S indicates systematic codes. Similarly, scheme PS80N has probability of W_1 selection of 0.8 with nonsystematic codes. It can be seen from the figure that the systematic codes generally have better results than the nonsystematic codes for the error range and data rates shown.

Systematic codes, in general, do not provide improvement compared to nonsystematic codes. Systematic codes however reduce the decoding complexity since with systematic codes the decoder operates with the matrix that has reduced number of rows (reduced, by the number of correctly received systematic packets). Thus, Figure 4 demonstrates that there is no loss in performance due to systematic codes.

Figure 5 shows PSNR versus PLR for the proposed systematic EW RLC scheme. The numbers shown in brackets represent the selection probability of each of the two windows, for example, UEP (60, 40) represents a code in which a symbol from W_1 will be selected for transmission with probability 0.60. As can be seen from the figure, the results of UEP schemes are significantly better than the EEP schemes for high loss rates.

UEP (100, 0) is a scheme in which only W_1 is protected and sent. The scheme is constrained in that it cannot achieve higher PSNR than 27.6 dB (see Table 1). However, the decoding failures, that is, when the entire GOP data fails to be decoded, will be much less for UEP (100, 0), since HPL is protected strongest which facilitates each GOP to be received with high probability, though at basic quality level. This scheme could thus prove useful in higher PLR. Also, note that, for this scheme, in Phase I of transmission, only the systematic codes in the HPL will be transmitted and, in Phase II, the encoded symbols come from HPL alone.

The PSNR results are improving with an increase in probability of selection of W_1 because at higher probabilities of selection of W_1 , the decoding of HPL has high chance to be successful. As described earlier, the PSNR with decoding of HPL is enhanced by systematic LPL packets.

In Table 2, the details of HPL size and PSNR contributions for the three schemes created with selecting three different values of T are shown. Intuitively, when the threshold T is lowered, the number of packets selected for HPL is higher. In Figure 6, we present the optimized results for the schemes created in Table 2. The results for the EEP scheme and benchmark are also shown for comparison. For each PLR, we found the optimal proposed UEP and the optimal benchmark UEP using (2). It can be seen from the figure that the proposed method leads to significant gains

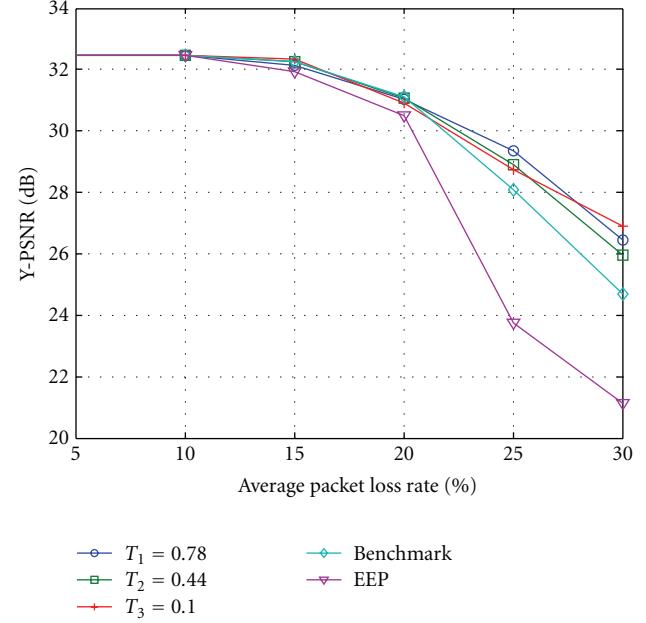


FIGURE 7: Optimized results for three values of T for the Paris sequence.

for high PLRs compared to the EEP and the benchmark scheme. The selection of T governs the size of HPL. If the size of selected HPL is small it will have relatively lower PSNR compared to a larger HPL. Lower T leading to a larger HPL, is thus better for higher PLRs, which is expected since a large HPL (with higher PSNR) is better protected, and for LPL anyway there is not enough bandwidth.

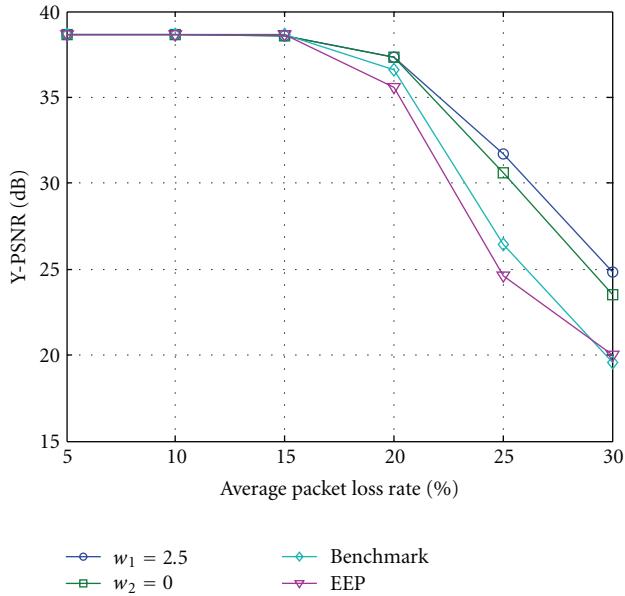
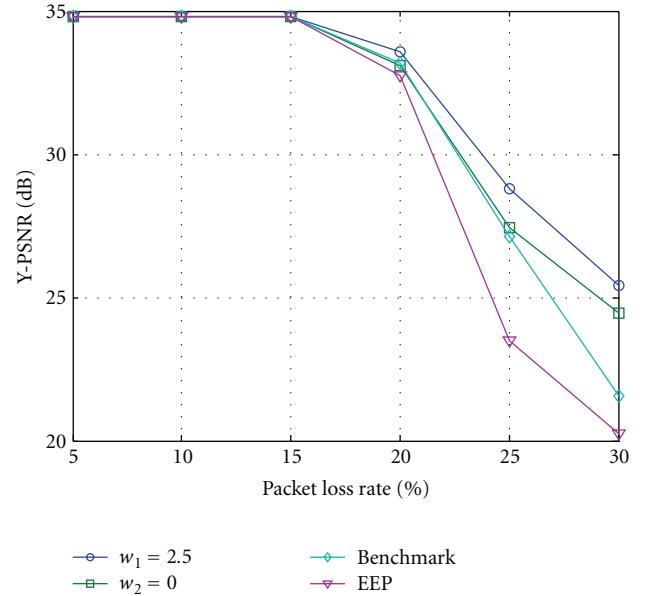
Similar results obtained for the CIF Paris video sequence are shown in Figure 7. Note that, for high PLR, it is better to reduce T resulting in large HPL. In any case, varying T , one can effectively design HPL/LPL sizes for different PLR.

A larger GOP size may be required for applications such as DVB-H [4]. We encode the same Foreman sequence with a GOP size of 64 frames. For this configuration, the total source packets are 161. The total number of sent packets is kept as 209 packets. In Figure 8, we present the optimized results for the schemes created using two different values of w as shown. Both schemes have the value of $T = 3.1$; however, based on different value of w , different slices are selected for HPL for each scheme. The scheme $w_1 = 2.5$ has better performance than $w_2 = 0$, especially at high packet loss, this comes from the fact that the former scheme prioritizes slices taking into account frame position in the sequence, which reduces error propagation. The benchmark scheme is created according to the selection criteria as used previously. EEP scheme performs the worst of all the schemes. The results for $w_1 = 2.5$ and $w_2 = 0$ are close at the lower PLR. The reason for this is that, with systematic codes, if the HPL is decodable, then the packets received correctly (which could be from HPL or LPL) in Phase-I also contribute to improve the PSNR.

The Paris sequence encoded with similar parameters is used to investigate the effect of w on performance. The

TABLE 2: Layer sizes and PSNR contributions for configurations with different values of T .

Layer	$T_1 = 0.78$			$T_2 = 0.58$			$T_3 = 0.44$		
	Size (bytes)	No. of packets	PSNR (dB)	Size (bytes)	No. of packets	PSNR (dB)	Size (bytes)	No. of packets	PSNR (dB)
HPL	21818	42	27.6	23598	45	28.25	25366	48	29.55
LPL	19218	35	36.39	17438	32	36.39	15670	29	36.39
Total	41036	77	36.39	41036	77	36.39	41036	77	36.39

FIGURE 8: Optimized results for two values of w for the Foreman sequence.FIGURE 9: Optimized results for two values of w for the Paris sequence.

optimized results are presented in Figure 9 for the schemes created using two different values of w along with Benchmark and EEP scheme. The results are similar to those in Figure 8 for the Foreman sequence, which confirms the analysis carried out earlier.

From the last two figures, we conclude that w is a useful parameter to improve source packet allocation (compared to the $w = 0$ case). We tested several different values of w and report our results for the several typical cases that show achievable performance boundaries by varying w . One can see from the figures that effect of w is small—up to 1 db.

5. Conclusions and Future Work

In this paper, we proposed the systematic EW RLC scheme to protect the sliced-partitioned video data under various channel conditions at different probabilities of window selection. We proposed a novel slice prioritization method that takes into account PSNR contribution of a slice as well as position of its frame within GOP. The simulations for two layers show that UEP schemes perform better as compared to the EEP scheme and ad hoc prioritization, achievable

with a minimal selection (one slice) of video data from each frame. Such reduced selections may be advantageously used in video-on-demand applications. The decoding complexity of RLC can be easily managed in the proposed scheme by an adaptive scheme which dynamically selects the slice size. The proposed schemes are hence suitable for real-time multimedia mobile applications.

Acknowledgment

This work has been accepted in part for presentation at IEEE ICON-2011 [29].

References

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [2] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

- [3] “ETSI TS 102 005, digital video broadcasting (dvb): specification for the use of video and audio coding in dvb services delivered directly over ip protocols,” ETSI Tech. Spec., 2006.
- [4] S. Nazir, D. Vukobratovic, and V. Stankovic, “Expanding window random linear codes for data partitioned H.264 video transmission over DVB-H network,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP-IEEE ’11)*, Brussels, Belgium, September 2011.
- [5] C. Hellge, E. Guinea Torre, D. Gmez-Barquero, T. Schierl, and T. Wiegand, “HDTV and 3DTV services over DVB-T2 using multiple PLPs with SVC and MVC,” in *Proceedings of the 61st Annual IEEE Broadcast Symposium*, Alexandria, Va, USA, October 2011.
- [6] L. Al-Jobouri, M. Fleury, and M. Ghanbari, “Raptor coding of H.264 data-partitioned video over a WiMAX channel,” in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE ’11)*, pp. 341–342, January 2011.
- [7] D. S. Lun, M. Médard, R. Koetter, and M. Effros, “On coding for reliable communication over packet networks,” *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.
- [8] J. Jin, B. Li, and T. Kong, “Is random network coding helpful in WiMAX?” in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08)*, pp. 191–195, Phoenix, Ariz, USA, April 2008.
- [9] D. Vukobratović and V. Stanković, “Unequal error protection random linear coding strategies for erasure channels,” *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1243–1252, May 2012.
- [10] “ETSI TS 126 346, universal mobile telecommunications system (umts); multimedia broadcast/multicast service (mbms); protocols and codecs,” ETSI Tech. Spec., 2005.
- [11] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [12] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, “Priority encoding transmission,” in *Proceedings of the IEEE FOCS*, pp. 604–612, Santa Fe, NM, USA, November 1994.
- [13] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, “Robust internet video transmission based on scalable coding and unequal error protection,” *Signal Processing*, vol. 15, no. 1, pp. 77–94, 1999.
- [14] Y. Lin, B. Liang, and B. Li, “Priority random linear codes in distributed storage systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1653–1667, 2009.
- [15] K. Nguyen, T. Nguyen, and S. C. Cheung, “Video streaming with network coding,” *Journal of Signal Processing Systems*, vol. 59, no. 3, pp. 319–333, 2010.
- [16] M. Halloush and H. Radha, “Network coding with multi-Generation mixing: analysis and applications for video communication,” in *Proceedings of the IEEE International Conference on Communications (ICC ’08)*, pp. 198–202, Beijing, China, May 2008.
- [17] X. Liu, G. Cheung, and C. N. Chuah, “Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast,” *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 730–741, 2009.
- [18] N. Thomos, J. Chakareski, and P. Frossard, “Randomized network coding for UEP video delivery in overlay networks,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’09)*, pp. 730–733, New York, NY, USA, July 2009.
- [19] S. Nazir, D. Vukobratovic, and V. Stankovic, “Performance evaluation of Raptor and Random Linear Codes for H.264/AVC video transmission over DVB-H networks,” in *Proceedings of the IEEE ICASSP*, Prague, Czech Republic, May 2011.
- [20] S. Nazir, D. Vukobratovic, and V. Stankovic, “Unequal error protection for data partitioned H.264/AVC video streaming with Raptor and Random Linear Codes for DVB-H networks,” in *Proceedings of the International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011.
- [21] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, “Scalable video multicast using expanding window fountain codes,” *IEEE Transactions on Multimedia*, vol. 11, pp. 1094–1104, 2009.
- [22] D. Vukobratović and V. Stanković, “Unequal error protection random linear coding for multimedia communications,” in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP ’10)*, pp. 280–285, Saint Malo, France, October 2010.
- [23] N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M. G. Strintzis, “Robust transmission of H.264/AVC streams using adaptive group slicing and unequal error protection,” *Eurasip Journal on Applied Signal Processing*, vol. 2006, Article ID 051502, 2006.
- [24] E. Baccaglini, T. Tillo, and G. Olmo, “Slice sorting for unequal loss protection of video streams,” *IEEE Signal Processing Letters*, vol. 15, pp. 581–584, 2008.
- [25] T. Tillo, E. Baccaglini, and G. Olmo, “Unequal protection of video data according to slice relevance,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1572–1582, 2011.
- [26] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.
- [27] ITU-T Recommendation H.264. Advanced video coding for generic audiovisual services. 2010.
- [28] H. Shojania and B. Li, “Random network coding on the iPhone: fact or fiction?” in *Proceedings of the 19th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV ’09)*, pp. 37–42, Williamsburg, Va, USA, June 2009.
- [29] S. Nazir, D. Vukobratovic, and V. Stankovic, “Scalable broadcasting of sliced H.264/AVC over DVB-H networks,” in *Proceedings of the IEEE International Conference on Networks (ICON ’11)*, Singapore, December 2011.
- [30] H.264/AVC Reference Software, <http://iphom.hhi.de/suehing/tm/>.

