Research Article

Rate Adaptive Selective Segment Assignment for Reliable Wireless Video Transmission

Sajid Nazir,¹ Dejan Vukobratović,² Vladimir Stanković,¹ and Ivan Andonović¹

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

² Department 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 2 December 2011; Revised 17 April 2012; Accepted 17 April 2012

Academic Editor: Sanjeev Mehrotra

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.

A reliable video communication system is proposed based on data partitioning feature of H.264/AVC, used to create a layered stream, and LT codes for erasure protection. The proposed scheme termed rate adaptive selective segment assignment (RASSA) is an adaptive low-complexity solution to varying channel conditions. The comparison of the results of the proposed scheme is also provided for slice-partitioned H.264/AVC data. Simulation results show competitiveness of the proposed scheme compared to optimized unequal and equal error protection solutions. The simulation results also demonstrate that a high visual quality video transmission can be maintained despite the adverse effect of varying channel conditions and the number of decoding failures can be reduced.

1. Introduction

Reliable real-time wireless video communication is gaining increased importance as novel richer multimedia applications are being deployed. Since wireless channels are prone to errors, it is necessary to provide strong error control mechanisms. Forward Error Correction (FEC) coding is preferable option as retransmission in real-time wireless applications is usually not a viable solution. On the other hand, error resilience video coding schemes generally come at a cost of decreased video performance in error-free environments and increased video coding complexity.

To combat packet drops, Digital Fountain erasure protection codes [1] are proven effective solution. Fountain codes [2, 3] are a recent class of FEC codes originally proposed for multicast/broadcast applications to combat losses of packets in the network. Fountain codes are rateless and in non-time-constraint applications can generate as many encoded packets as needed. The amount of additional packets transmitted is the redundancy that is necessary for decoding to succeed and can be adjusted to combat different channel conditions. In bandwidth-limited wireless networks it is important to keep the introduced redundancy to a minimum. Thus, instead of targeting the worst possible channel conditions,

the redundancy should be adaptively adjusted according to the varying channel conditions via dynamic source-channel coding. LT codes [2] are the first proposed class of practical fountain codes. Although Raptor codes [3] generally provide better performance, the LT codes are used in this paper due to their design and implementation simplicity. Note, however, that LT codes have a higher decoding complexity of $O(k \log_e k)$ per source message (where k is the message length) than Raptor codes, $O(k)$.

H.264 Advanced Video Coding (AVC) [4] is the state-of-the-art video coding standard achieving significant compression efficiency and gaining widespread use in the emerging communications standards and applications. When transmitting H.264/AVC video over a wireless channel, due to significant fluctuations of channel characteristics, the video is encoded at a fixed source rate and the redundancy (i.e., LT coded symbols) is added to avoid error effects. Usually, for simplicity, the entire video block is protected equally using equal error protection (EEP).

An alternative is to classify the encoded content based on the importance to the reconstruction and assign different amount of redundancy to different importance classes using unequal error protection (UEP). For example, intracoded frames can be protected stronger than inter-coded ones.

Another option is to use a higher source coding rate and to continuously adapt the source rate to the varying channel bandwidth by dropping some of the frames, in order to keep the channel coding rate low enough. This joint source channel coding option can be combined with EEP to lead a simple rate adaptive solution or with UEP to provide a more complex, but optimized protection.

In this paper we proposed Rate adaptive selective segment assignment (RASSA) scheme and compare its performance to that of fixed source rate EEP and fixed source rate UEP. We resort to error resilience and concealment features, designed to make the video less vulnerable to the effects of lost data, and then compress the video at a higher source rate allowing for some decoding errors. In particular, in this paper, we study data partitioning and slicing. Data partitioning (DP) [4] is a low-cost error resilience feature, supported by the extended AVC profile, which can be exploited to introduce a layered structure in H.264/AVC. The DP feature of H.264/AVC effectively prioritizes a video stream by partitioning it into classes of different importance to video reconstruction with a very small rate penalty compared to the AVC standard without error resilience.

Besides DP, it is possible to partition a frame into a fixed number of *slices*, which are of different importance to the video reconstruction. Thus, similarly to DPs, the slices can be aggregated into different priority classes, with the higher-priority classes containing slices that have higher contribution to the reconstruction. Such prioritization can make the sliced video data amenable to UEP and rate adaptation.

A lot of work has been done on joint source-channel coding; see [5] for a review. In the domain of rateless source channel coding, in [6], a class of unequal error protection codes, called Expanding Window Fountain (EWF) codes, is used for UEP of scalable video. In [7], unequal protection has been proposed for video communications by duplicating the information symbols and extending the original LT degree distribution to the new set of information symbols. In [8], unequal Growth codes have been proposed where as the number of packets the receiver has increases, the degree for each new encoding symbol needs to increase, hence the name Growth codes. An adaptive rateless coding for DP AVC coded video has been proposed in [9]. The proposed system uses intracoded macroblocks (MBs) in each frame; some additional redundant data is piggybacked onto the ongoing packet stream. In contrast, this study uses an IPPP... structure, where each GOP is treated as a source block for LT coding. The contributions of this study are (1) analysis of optimized EEP and UEP schemes for transmission of DP and sliced H.264/AVC video and their robustness in channel mismatch scenarios and (2) a rate-adaptive optimized solution for bandwidth-limited wireless channels and limited resource devices.

Although LT codes are used in the simulation section, the proposed solution can be applied to other rateless packet loss protection codes. A scheme has been proposed to decode video even when the rateless decoding fails using packetization information. This is made possible by passing

a videotable to the decoder containing the DP type and size information. Thus, the DPs with all or part of their data missing are discarded before the H.264 decoder tries to decode the data. It is important to note that without such information the decoding will fail on encountering such missing data.

The segmentation of video data facilitates a layered coded video that might be preferable to the H.264 Scalable Video Coding (SVC) extension [10] in some applications since it complies with the AVC standard and provides scalability, and more robust output to packet losses than SVC. The proposed scheme can be applied in multicast scenarios with heterogeneous receivers, in which case a receiver can terminate reception and decoding of segments after having received data compatible with its processing power and memory.

The rest of the paper is organized as follows. Section 2 covers the background of DP, slicing, and LT erasure protection coding. In Sections 3 and 4, the proposed system and the proposed rate allocation algorithms are described, respectively. The results and analysis are in Section 5. Finally, the conclusion and future research directions are contained in Section 6.

2. Background

In this section we give background on error resilient H.264/AVC and erasure protection coding used. H.264/AVC formats the video data into Network Abstraction Layer (NAL) units enabling it to be transported over various channels. Each video frame is encapsulated in a separate NAL unit. H.264/AVC provides many errorresilience options to mitigate the effect of lost packets during transmission. Next, we briefly outline two options used in this paper.

2.1. Data Partitioning. A low-cost option is the DP [4, 11] which supports the partitioning of a frame/slice in up to three partitions (NAL units), based on the importance of the encoded video syntax elements for video reconstruction (see Figure 1(a)). DP *A* contains the most important data comprising slice headers, quantization parameters, and motion vectors. DP *B* contains the intracoded macroblocks (MB) residual data, and DP *C* contains inter-coded MB residual data. This importance-based partitioning enables assigning different protection levels to different partitions. The decoding of DP *A* is always independent of DP *B* and *C*. However, if DP *A* is lost the remaining partitions cannot be utilized. To make decoding of DP *B* independent of DP *C*, Constrained Intra Prediction (CIP) parameter in the H.264/AVC encoder is set. The loss of an NAL unit can result in error propagation to later frames due to interframe dependence.

2.2. Slicing. Another scheme available in the baseline profile is slicing [4], 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 MBs or fixed data rate. Each coded slice is independently

FIGURE 1: (a) Data partitions. (b) Segmented data partitions.

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.

DP has low overhead as its structure is determined in advance, whereas slicing generally requires a slice group map.

2.3. LT Codes. The first practical class of fountain codes are LT codes [2]. The LT encoder can potentially generate an unlimited number of encoded symbols from a limited set of source symbols. Encoded symbol is obtained by selecting uniformly at random d different source symbols and their bitwise XOR-ing. The degree d of each encoded symbol is drawn i.i.d. from a discrete probability distribution $\Omega(d)$ called the degree distribution. LT codes designed using Robust Soliton degree distribution are asymptotically capacity-achieving in combination with the iterative Belief-Propagation LT decoder [2]. It may be worth noting that many implementations of rateless coding, such as systematic 3GPP Raptor code [12], do not use the belief propagation algorithm but employ matrix operations instead.

The LT decoder gathers the received encoded symbols and tries to recover the original source symbols. The decoder needs to know the degree and the location of source symbols, which have been combined together to form the encoded symbol. The decoder keeps on processing the encoded symbols of degree one, recovering a source symbol that is then XOR-ed with all the symbols it is connected to and the corresponding LT code graph edges are removed. This process continues until the decoding succeeds or stops with errors [2].

2.4. UEP Schemes. In order to enable UEP, the video data is divided into two segments/layers according to its importance for video reconstruction. Intuitively, we put the important data, that is, IDR and DP A, always in so-called high-priority layer (HPL). The other layer termed low-priority layer (LPL) contains least important data.

The UEP schemes are based on varying the probability of selection of HPL. Note that the same rateless codes are used for protection of both HPL and LPL, and UEP is achieved by probabilistically selecting at the transmitter for each output symbol whether it should come from the HPL or LPL stream. Thus, instead of two different fixed code rates, we use soft code rates via defined selection probability of HPL. If we increase the selection probability of HPL, we improve its robustness at the price of a decreased robustness of LPL. Also, it is important to take into account the relative sizes of the priority layers. The selection probability of a layer must

at least correspond to its relative size. Moreover, assigning a higher selection probability than required to HPL could be beneficial in cases where more protection to it is required.

3. The Proposed System

In this section we describe the proposed system that segments encoded video and provides equal or unequal error protection. First, we describe a system that forms a layered output using the DP feature. Then, we present the system that exploits slicing instead of DP.

3.1. Protection of DP-Based AVC Video. The video data of each non-IDR frame is divided into three data partitions by the H.264/AVC encoder. IDR frames were not partitioned and they were put always into the HPL. This partitioned data needs to be aggregated together to enable UEP. The structure of a segmented video is shown in Figure 1. The figure shows the DP A, B, and C together with the I frame. (Note that the first non-I frame is denoted as A_1, B_1, C_1 , and so forth.) Next, we prioritize the partitions and group all DP As, Bs, and Cs together effectively forming three segments or layers as shown in Figure 1(b). Note that by receiving only the I/Instantaneous decoder refresh (IDR) frame and DP $A_1, \dots, A_n$, the decoder will still be able to decode all n frames within the group of pictures (GOP), though at reduced quality. Further segmentation is not restricted to be done at the aggregate partition boundaries only. That is, if all IDR and DP A are sent as the first segment, then any number of DP B and DP C partitions can be selected for transmission in the second segment. It is worth noting that this will only work for pre-encoded video.

This gives flexibility that enables a fine-grained layered structure as a large number of reconstruction rate points become available, which can be matched to the channel statistics with a very fine control over video reconstruction quality. The DP B and DP C by virtue of having been aggregated are already in their priority order for reconstruction. The layer with important data (IDR and DP A...) is termed HPL, whereas the remaining data is placed in LPL. In the proposed scheme, intra-refresh MBs are not used but instead periodic I frames are assumed.

The segmented data partitions are next protected by FEC codes applied on each GOP independently.

To achieve UEP, each segment should be protected according to its importance using different amount of redundant symbols. The symbol size is 70 bytes. To accomplish that, the FEC encoding process adds an important initial step, that is, to first select a segment from which the encoded symbol is to be generated determined by "selection

probability” of a segment, which is a preassigned parameter based on the importance of different segments and the data rate available. After a segment is selected, a conventional encoding is performed over the source packets contained in that particular segment only. Thus, instead of defining a UEP scheme as a set of rates (one for each segment), we equivalently define it by a set of selection probabilities. This resembles the method of [6]. For practical reasons the number of layers in the UEP is usually constrained to two or three.

Note that the UEP scheme allocates redundancies to the segments based on their importance. The optimal rate allocation depends not only on the channel characteristics but also on video data since the importance and sizes of the segments vary from a GOP to a GOP. Thus, the UEP has to be dynamically changed and optimal allocation needs to be found for each GOP, which is practically feasible only for a prerecorded video. Note that in the extreme case when the bandwidth is very scarce or packet loss rate high, which is often in mobile wireless scenarios, the optimal selection probability of low-priority segments would be zero and all redundancy would be allocated to the high-priority segments to ensure their successful decoding.

Motivated by this and targeting wireless applications with limited bandwidth available and high loss rates, we introduce another scheme called the RASSA scheme. The RASSA scheme is a special case of UEP that exploits the flexibility of layered coding of DP and slicing. First, given an estimated packet loss rate and total rate budget, the system calculates the required overhead (and thus also the amount of source data) that will allow for error-free transfer with high probability (w.h.p.). Then, the data is filled starting from leftmost in Figure 1(b), and remaining source data is discarded. This way, the scheme discards some of the lower-priority data by assigning zero selection probability, to increase protection of the more important data.

Thus, this scheme is not constrained in having two or three segments/layers, and any number of DPs/slices can be selected enabling a very flexible rate control. For example, given channel statistics, we can provide enough redundancy for a segment containing DP A and B and part of DP C to be recovered at the decoder w.h.p. The unselected low-priority data (remaining DP Cs) are simply discarded. Note that either the entire sent source block will be decoded, or decoding will fail, in which case the previous GOP is used for reconstruction.

RASSA can be seen as a UEP scheme since it protects only one part of the encoded data and discards the rest, but also as EEP since it provides equal protection of all sent source data. One immediate advantage of this scheme is reduced complexity since only one code is used, where UEP generally requires one code for each layer, and there is no need for complex rate optimization. Indeed, once the channel loss rate is estimated, the required code rate is set, and based on the available bandwidth (total budget) the decision to drop some of the NAL units that cannot fit the total budget is made.

UEP schemes require that the DPs of each type in LPL are aggregated together. To pass this information to the decoder, we propose a video table structure to be created at

the encoder. The encoded video generated by the H.264/AVC encoder with DP is used to create a video-table with an entry for each NAL unit and its length. The number of NAL units per GOP is usually small (up to 64), and hence the table can conveniently be passed to the decoder within a header with negligible rate increase. The packet bearing header will be transported with the HPL. If HPL is lost, then anyway no video decoding is possible.

At the receiver side, the video-table structure is used to rearrange the DPs to their original encoding order. The table is also used to discard NAL units with missing data. That is, since one DP/NAL can be sent in multiple packets, if one packet is missing the entire DP is dropped. Also, recovered DP B and DP C of a frame are dropped if DP A for that frame is not recovered properly.

There is negligible latency involved in bringing the DPs to their original order for decoding. The aggregation of DPs is only limited to a priority layer. For instance, if DP A and DP B both are in HPL, then they will remain in their original encoding order.

3.2. Protection of Sliced AVC Video. In our previous work [13], we propose and test a method for segmenting sliced-AVC output into multiple segments based on importance of the slices to reconstruction. For example, we can form two priority classes where more important slices, which contribute to the peak signal-to-noise ratio (PSNR) level above a fixed threshold, are put in the HPL, and all other in LPL. Then, the protection methods described above (EEP, UEP, and RASSA) can be applied to such prioritized data without modification.

We encode a video sequence using slicing with each frame divided into a fixed number of slices. The priority of each slice is obtained by dropping it from the GOP data and measuring the resulting PSNR, as a frame-by-frame average of the entire GOP, by actual decoding. In view of the encoding latency, the scheme is meant for pre-encoded video. 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. The importance of the slices on total frame-averaged PSNR generally decreases as we move towards the end of the GOP. 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. Details of assigning slices to different layers can be found in [13].

4. Rate Allocation

In this section we discuss rate allocation optimization for the three proposed schemes. We assume that DP is done; however, in the same way, rate allocation can be done in case of slicing.

Let N be a given total rate budget expressed as the total number of packets/symbols that can be transmitted for each GOP. The video is encoded using DP H.264/AVC forming either four segments, IDR, DP A, DP B, and DP C, or two classes of slices. We assume that each segment can be truncated arbitrarily. Let K be the total number of encoded source packets/symbols.

We consider three schemes: (i) an EEP scheme that generates N packets using all K source packets and transmit them over the network; (ii) a UEP scheme that groups source data into L importance layers starting from IDR; for example, we can have $L = 4$ where each of four segments forms one layer; (iii) an RASSA scheme that takes first $K_{\text{RASSA}} \leq K$ source packets to generate N transmission packets.

Assuming that video is pre-encoded, K is fixed and is not part of the optimization. Then, the EEP scheme always uses an (N, K) code and thus does not require optimization.

An L -layer UEP scheme can be described by L -tuples $\pi = (p_1, p_2, \dots, p_L)$ and $k = (k_1, k_2, \dots, k_L)$, where p_i and k_i represent the selection probability and the size in packets, respectively, of layer i . Then the optimal rate allocation between the L layers can be found by maximizing the expected PSNR of the reconstruction given by

$$\widehat{\text{PSNR}} = \sum_{i=0}^L P_i(\pi, K) \text{PSNR}_i, \quad (1)$$

where P_0 is the probability that no layer is recovered, P_i is the probability that first i layers can be recovered but not layer $i + 1$, and P_L is the probability that all layers can be recovered successfully. The task is to find L -tuples π^* and K^* that maximize the expected PSNR, over all possible L -tuples π and K . P_i can be obtained experimentally or for some FEC codes estimated analytically for each π , K and each channel condition and are source independent. For simplicity, it is assumed that K is set a priori by the video encoder which is often the case. Indeed, it is natural to group all packets from one segment together. For example, for $L = 3$, IDR and DP A can be placed into one layer, DP B in another, and DP C in the last layer. Note that the sizes of each segment are determined by the video encoder, and are not subject of the optimization. The problem can further be simplified by maximizing the expected received rate instead of PSNR as

$$\hat{R} = \sum_{i=0}^L P_i(\pi) K_i, \quad (2)$$

where K_i is the number of packets in the first i layers and $K_0 = 0$. This way, the optimization is independent of the source content and depends only on the total rate, layer sizes, and channel loss rate. There are many methods proposed to efficiently accomplish the two optimization tasks (see [5, 6] and references therein).

For the RASSA scheme, recall that out of K generated source packets, only K_{RASSA} are selected that are protected by an (N, K_{RASSA}) channel code before transmission. The optimization problem is simplified to the following. Given a total number of transmission packets N and packet loss rate q , the task is to find the number of sent source packets

$K_{\text{RASSA}} \leq K$, such that all K_{RASSA} source packets can be decoded w.h.p. Note that determining K_{RASSA} implies the used channel code (N, K_{RASSA}) . Again, the expected PSNR or the expected number of received source packets is maximized, given by

$$\begin{aligned} \widehat{\text{PSNR}} &= (1 - P) \text{PSNR}_0 + P \cdot \text{PSNR}_1, \\ \hat{R} &= P \cdot K_1, \end{aligned} \quad (3)$$

respectively, where P is the probability of successful decoding and PSNR_0 and PSNR_1 are reconstructed PSNR if decoding fails or is successful, respectively. K_1 denotes the number of source packets sent by the RASSA scheme. Note that P depends on q and K_1 and can be found experimentally or analytically. Indeed, for maximum distance separable codes, P is the probability that the number received packets is at least K_1 , and then

$$\hat{R} = \binom{N}{K_1} q^{N-K_1} (1-q)^{K_1} K_1, \quad (4)$$

which can be solved numerically.

In the next section, we will compare results of the rate and PSNR-optimized RASSA schemes to that of EEP and optimized UEP schemes.

5. Results and Analysis

We test robustness of the EEP, optimized UEP, and RASSA schemes when packet loss rates q and data rates N vary. We show effectiveness of the proposed approach using both DP and slicing features. Simulations have been performed using the H.264/AVC software JM 16.2 [14]. A GOP size of 16 frames is used with the IPPPP... structure. We report results for two video sequences “Paris” and “Football.” Penalty due to DP and slicing for these sequences is up to 0.1 dB.

5.1. DP AVC Transmission. We assume that the video has been pre-encoded at a fixed rate using DP into fixed length segments IDR, DP A, DP B, and DP C. The data in each segment is formed into source symbols/packets of size 70 bytes for the LT coding process, which is a good compromise between performance and complexity. The IDR frame is put in the first NAL unit and it is not partitioned. CIP is used to make the decoding of DP B independent of DP C. Each non-IDR frame is partitioned into DP A, DP B, and DP C.

The partitions with their relative sizes and the PSNR contribution for the first GOP of the CIF format “Paris” and “Football” video sequences are shown in Table 1. We consider a two-layer UEP scheme where the first, HPL, contains selected more important partitions, and the second, LPL, contains the remaining partitions. The UEP schemes are described by $\text{UEP}(p_1, p_2)$, where p_1 and p_2 represent the selection probabilities of packets from HPL and LPL, respectively, and the optimal solution can be found as shown in Section 4.

TABLE 1: Partition sizes for the first GOP of “Paris” and “Football” sequences.

Partition	Paris		Football	
	Size	Cum. PSNR	Size	Cum. PSNR
IDR	22281	—	23374	—
DP A	12838	30.13	22823	—
DP B	97	30.32	2893	25.39
DP C	45732	39.16	31731	32.62
Total	80948	39.16	80821	32.62

In Tables 2 and 3 we show the classification of the DPs and the resulting LT packets, for the “Paris” and “Football” sequences, respectively.

After FEC coding, one encoded symbol (together with RTP/UDP/IP headers) is placed in an IP packet and is subjected to a uniform and Gilbert loss pattern with average loss rates of 5, 10, 15, and 20%. For the Gilbert model the average burst length is 5. We assume use of header compression, and thus a 4-byte header is considered. The base data rate is set to 1000 kbps, and the successively higher rates are obtained by adding roughly 10% additional symbols, up to a rate 1.5 times higher than the base rate.

The simulations are performed using one slice per frame and a frame rate of 25 frames per second (fps). The selected schemes are simulated with 100 runs for each GOP. In cases where the entire GOP is lost, the PSNR is obtained using the last frame of the previously decoded GOP to replace all frames of the lost GOP.

We report results for the EEP and UEP schemes and compare them to the results obtained with two optimized RASSA schemes: SS-PSNR and SS-Rate schemes. The results with frame-by-frame average PSNR performance of the five selected configurations at 1.1Mbps for selected packet loss rates are shown in Figures 2 and 3, for the “Paris” and “Football” sequences, respectively. “Opt-UEP” denotes the scheme that is optimized for each packet loss rate. As can be seen from Figure 3, the performance of the EEP scheme is the worst. The performance of the UEP schemes gets better with an increase in the protection of HPL. UEP(60,40) performs worse as compared to UEP(80,20) because the protection gets divided over both segments and none is protected enough. SS-PSNR performs the best of all the schemes. For the “Football” sequence performance of the optimized UEP scheme is very close to that of the SS-PSNR. Similar results with the same parameters as Figure 2 are shown in Figure 4 for burst loss.

The results showing PSNR performance of the five selected configurations at 10% packet loss rates for different data rates are shown in Figures 5 and 6, for the “Paris” and “Football” sequences, respectively. The performance of the EEP scheme gets progressively better at higher data rates. SS-PSNR and SS-Rate provide reliable and consistent performance at all the data rates. UEP(80,20) is limited to 30 dB in Figure 5 even at higher rates because the DP C is not getting enough protection. Interestingly, at the highest rate the EEP scheme is better than the optimized UEP scheme,

TABLE 2: Priority class and LT packetization for first GOP of “Paris” sequence.

Class	DP	PSNR	Number of bytes	Number of LT packets
HPL	IDR + DP A	30.13	35119	502
LPL	DP B + DP C	39.16	45829	655
Total	80,948	39.16	80948	1157

TABLE 3: Priority class and LT packetization for first GOP of “Football” sequence.

Class	DP	PSNR	Number of bytes	Number of LT packets
HPL	IDR + DP A + DP B	25.39	49090	702
LPL	DP C	32.62	31731	453
Total	80,821	32.62	80821	1155

due to the absence of the performance penalty introduced by DP. In Figure 7 the results for the burst channel model are given.

6. Sliced AVC Transmission

In this section we present our simulation results with the slicing feature. For simplicity, we consider the case of $L = 2$ layers: HPL that contains more important slices and LPL that contains less important slices [13]. The same video parameters are used as in the previous subsection.

The sizes, number of packets, and resulting PSNR values for the “Paris” video sequence are shown in Table 4.

The results are shown in Figures 8 and 9 and confirm the analysis carried out with the DP schemes. SS-PSNR is the best scheme overall. The UEP schemes, except UEP(45,55) in Figure 9, are around 24 dB as they suffer from an overprotection of HPL. This is because the HPL size is only about 43% of the GOP size. This highlights the significance of considering the HPL size while designing UEP schemes. The EEP scheme becomes better than the UEP schemes at high data rates. Figure 10 shows the results for the burst loss model. Similar results are obtained for the “Football” sequence.

7. Discussion and Future Work

Although both DP and slicing have been demonstrated to enable efficient layered video data transmission, the results with DP are seen to be better. The sizes and number of DPs generated are as determined by the encoder. The prioritization of data into various partitions is thus optimum and can easily be used to create different rate points. Slicing, on the other hand, is more flexible as it allows for a finer layered structure. Moreover, in contrast to DP, slicing is available in the baseline AVC profile. However, simulation results show small advantage of the DP-based

FIGURE 2: PSNR versus PLR at overall data rate of 1.1 Mbps for the “Paris” sequence—uniform loss.

FIGURE 3: PSNR versus PLR at overall data rate of 1.1 Mbps for the “Football” sequence—Uniform Loss.

scheme compared to the slicing-based, especially at high packet loss rates.

The performance of different coding schemes with segmented H.264/AVC video data has been analyzed. The segmented data can be selected to suit the available data rate and channel conditions. The UEP schemes provide better performance over EEP at some rates only. The RASSA scheme can be used to match the available transmission video data to the instantaneous channel conditions. It combines the best of both the EEP and UEP schemes to provide

FIGURE 4: PSNR versus PLR at overall data rate of 1.1 Mbps for the “Paris” sequence—burst loss.

FIGURE 5: PSNR versus data rate at PLR of 10% for the “Paris” sequence—uniform loss.

better and reliable video quality even in the worst channel conditions. The passing of the video-table to the decoder is a low-cost solution to an “all or nothing” decoding. Note that it is assumed that the video is pre-encoded, and thus the best way to match the source rate with the channel rate is to selectively drop some of the DPs, which is done in RASSA. Indeed, the results presented here show that the pure UEP with fixed source rate suffers huge performance loss compared to the scheme that adjusts the source rate. The main advantage of the proposed scheme is a very simple adaptation of the source rate via DP AVC coding. Note that RASSA can be combined with UEP to better match source and channel characteristics. However, that

FIGURE 6: PSNR versus data rate at PLR of 10% for the “Football” sequence—uniform loss.

FIGURE 7: PSNR versus data rate at PLR of 10% for the “Paris” sequence—burst loss.

TABLE 4: Priority class and LT packetization for first GOP of “Paris” sequence.

Class	Size (bytes)	Cum. PSNR	Number of LT packets
HPL	34779	24.3	497
LPL	45536	39.08	651
Total	80315	39.08	1148

would require multiple channel codes, increased complexity, UEP optimization algorithms, and reduction of the channel code length used could worsen channel codes’ correction capabilities. This will be part of future work by incorporating expanding window codes [6].

FIGURE 8: PSNR versus PLR at overall data rate of 1.1 Mbps for “Paris” sequence—uniform loss.

FIGURE 9: PSNR versus Data rate at PLR of 10% for the “Paris” sequence—uniform loss.

The combined use of FEC and adaptively dropping some DPs to maximize PSNR is thus shown as a practical method to ensure reliable delivery of multimedia data over wireless channels.

FIGURE 10: PSNR versus data rate at PLR of 10% for the “Paris” sequence—burst loss.

Acknowledgment

D. Vukobratović was supported by a Marie Curie European Reintegration Grant FP7-PEOPLE-ERG-2010 “MMCODE-STREAM” within the 7th European Community Framework Programme.

References

- [1] M. Mitzenmacher, “Digital fountains: a survey and look forward,” in *Proceedings of the IEEE Information Theory Workshop (ITW '04)*, pp. 271–276, San Antonio, Tex, USA, October 2004.
- [2] M. Luby, “LT codes,” in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.
- [3] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [4] S. Wenger, “H.264/AVC over IP,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [5] R. Hamzaoui, V. Stanković, and Z. Xiong, “Optimized error protection of scalable image bit streams,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 91–107, 2005.
- [6] 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, no. 6, pp. 1094–1104, 2009.
- [7] S. Ahmad, R. Hamzaoui, and M. M. Al-Akaidi, “Unequal error protection using fountain codes with applications to video communication,” *IEEE Transactions on Multimedia*, vol. 13, no. 1, pp. 92–101, 2011.
- [8] A. G. Dimakis, J. Wang, and K. Ramchandran, “Unequal growth codes: intermediate performance and unequal error protection for video streaming,” in *Proceedings of the 9th IEEE International Workshop on Multimedia Signal Processing (MMSP '07)*, pp. 107–110, Crete, Greece, October 2007.
- [9] L. Al-Jobouri, M. Fleury, and M. Ghanbari, “Adaptive rateless coding for data-partitioned video streaming over a broadband wireless channel,” in *Proceedings of the 6th Conference on Wireless Advanced (WiAD '10)*, pp. 1–6, June 2010.
- [10] 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.
- [11] R. Razavi, M. Fleury, M. Altaf, H. Sammak, and M. Ghanbari, “H.264 video streaming with data-partitioning and growth codes,” in *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP '09)*, pp. 909–912, Cairo, Egypt, October 2009.
- [12] ETSI Technical Specification, “Universal mobile telecommunications system (umts); multimedia broadcast/multicast service (mbms); protocols and codecs,” ETSI TS 126 346, 2005.
- [13] S. Nazir, D. Vukobratovic, and V. Stankovic, “Scalable broadcasting of sliced H.264/AVC over DVB-H network,” in *Proceedings of the IEEE International Conference on Networks, Special Session on Robust and Scalable Multimedia Networking (ICON '11)*, Singapore, December 2011.
- [14] H.264/AVC Reference Software, <http://iphome.hhi.de/suehring/tml/>.