Robust Video Communication for Ubiquitous Network Access

Hannadi Samek[1] **· Martin Fleury** [2] **· Mohammed Ghanbari**[2]

[1]*University of Aleppo, Aleppo, Syria*
dr_hanadii@yahoo.com
[2]*University of Essex, Colchester CO4 3SQ, United Kingdom*
{fleum, ghan}@essex.ac.uk

Abstract Ubiquitous network access implies that video can be streamed to portable devices whether they are moving outdoors or docked at home. Unfortunately, broadband wireless channels and their wired alternatives present a hostile environment for video communication, which manifests itself in error bursts. This paper presents a robust application-layer, channel-coding scheme suitable for data-partitioned, compressed video. Data partitioning prioritizes the more important data within a compressed bitstream. In the scheme, the more important compressed data are protected prior to communication over an access network. In particular, window-growth rateless codes are used. This form of rateless code can be incrementally scaled to reflect the importance of the data being protected. The paper gives details of the scheme for achieving this in the context of an H.264/AVC codec's picture types and structures. The paper considers how best to apply the scheme to H.264/AVC's data-partitioning modes in a practical manner. Simulations of error-prone channels show that the proposed unequal protection scheme achieves several dBs of improvement in video quality, when compared with equal protection. The simulations modeled both wireless and wired access networks.

Keywords data-partitioned video · FEC· rateless channel coding · Window growth codes · video streaming

1 Introduction

The ability to stream video to and from portable devices is a common requirement in ubiquitous computing environments. For example, work in [19] analyzed the problems of providing a video-on-demand service to devices in a ubiquitous environment with differing bandwidth capacities and storage capabilities. However, there is a further issue with video communication and that is the fragile nature of a compressed video bitstream. Due to data dependences within the stream, loss of data can have a damaging effect on video quality, though the loss does not always prevent the decoder attempting a reconstruction. Video streaming remains an intimate part of a ubiquitous environment. For example, in [23], video is one of the modalities for mobile geo-blogging, and if the video blogs are to be conveyed to others, video will need to be streamed, as storage on portable devices is limited. In [22] the social context of video recording is explored. For these applications and augmented reality as well, the degree of protection offered the video stream impacts upon the video quality of experience.

Transmission of video over error-prone channels demands better protection of the compressed bitstream in order to ensure an acceptable quality-of-service (QoS). Some protection schemes for error-prone channels employ Automatic Repeat reQuest (ARQ) to retransmit previously corrupted data. Others may first probe the channel in order to adjust the number of retransmissions [31] according to transmission conditions. The research in [12] proposed a scheme for wireless channels that varied the extent of selective repeat ARQs according to both the channel conditions and to the relative energy budget for Reed-Solomon (RS) coding. However, the volatility of such channels may make estimates unreliable. Consequently, ARQ may not perform well. In addition, retransmission incurs delays, which can cause streaming applications to miss display deadlines. Nevertheless, it is actually possible to protect video against errors without retransmission if layered coding [8] takes place. For example, by combining layered video coding with Forward Error Correction (FEC), the degree of protection of higher priority layers can be increased relative to the degree of protection for less important lower layers.

One form of layered coding is through data-partitioning (DP) [25]. In DP, the compressed video stream is partitioned according to the data priority. Therefore, each partition priority forms a coding layer. Prioritization is in terms of the data's contribution to the reconstruction of the video. In this paper, we combine application-layer FEC in the form of rateless channel coding with DP to protect transmission in error-prone channels.

Compressed video to some extent can withstand errors and, unlike alpha-numeric data, does not need to be perfectly reconstructed. Decoding is still possible if essential data such as motion vectors (MVs) arrive intact. Consequently, the family of rateless or Fountain codes [16] is an attractive option for the protection of video against channel errors. In this channel coding method, a varying degree of redundancy is incrementally added to a group of symbols, to ensure that the symbols can be decoded under a variety of adverse channel conditions. Thus, unlike RS codes, the coding rate is not fixed at the time of coding but can be dynamically varied. The degree of redundancy depends on the severity of the channel conditions. After a feedback request, symbols can be gradually transmitted to the receiver, until the delay limit is exhausted.

Consequently, rateless codes are now attracting applications in video streaming. For example, in [2] rateless coding was applied to packets in unicast video streaming over the Internet. In [20] rateless coding was selected for reasons of reduced decode computational complexity in an energy reduction scheme for wireless mesh networks.

In this paper, we newly apply a window-growth rateless code for data-partitioned video that can provide good video quality at a small decoding delay. Window-growth codes [29] are an extension of rateless codes, which allow the amount of protection to be incrementally scaled. As such they can allow prioritized protection of the more important of the partitions. This paper demonstrates an innovation in the application of window-growth rateless coding for DP, which is able to lessen transmission delay with reduced redundant overhead.

In fact, the main contribution of the current paper is an analysis of how precisely the scheme can be applied in a state-of-the-art H.264/AVC (Advanced Video Coding) codec [27]. In this codec, there is a complex arrangement for DP according to picture type, DP mode, and H.264/AVC profile.

As a demonstration of the proposed scheme, experiments have been applied to a wireless channel subject to burst errors. Such errors frequently occur due to slow and fast fading. We also include additional experiments on an Asymmetric Digital Subscriber Line (ADSL) twisted-pair channel [4], which is also subject to 'bursty' errors. ADSL is, of course, the principal wired alternative to wireless access of the Internet.

Mobile TV is set to become a ubiquitous service available over a variety of networks. TV services are extending beyond the traditional terrestrial and satellite broadcast forms to mobile varieties such as Digital Video Broadcasting-Handheld (DVB-H) and Digital Multimedia Broadcast (DMB) [11]. Within the network sphere, IPTV provides services such as live TV, time-shifted TV, and video-on-demand. Hence, IPTV (Internet Protocol TV) [18] is the most likely application of the proposed scheme.

However, 'over-the-top' TV (IPTV over broadband) suffers from error-prone channels, whether the final delivery step is over broadband wireless access or ADSL. In particular, the video community has considered [13] the effect of error bursts on compressed video bit-streams. Because of the predictive nature of video, source-coded video is particularly sensitive to this type of multiple error. As video must be delivered at video frame rates, when TV arrives from the content distribution network source, it will already have been delayed over the network path.

Hence, the significance of the FEC scheme developed in this paper, as the scheme reduces the need for retransmissions, even though some retransmissions are still required.

The rest of this paper is organized as follows. Section 2 comprises background material on the organization and structure of an H.264/AVC codec's compressed output and on rateless codes in general. Section 3 shows how window-growth codes can be applied to a data-partitioned, compressed video bitstream. The Section details the scheme for all data-partitioned types. This includes the way retransmission of redundant data takes advantage of non-essential data in the bitstream to reduce the number of retransmissions. Section 4 is a simulation to demonstrate the effect of the scheme, while Section 5 draws some conclusions.

2 Background

This Section reviews the background information necessary for an understanding of the scheme.

2.1 Video coding essentials

Standard video codecs [9], for computational convenience, decompose the decoding process. A 16 × 16 pixel block known as a macroblock (MB) is the smallest coding unit. A slice is a collection of MBs within a picture formed in support of error resilience. Within a slice, an Intra-MB is independently coded without reference to the MBs of previous pictures, though it may be spatially predicted within the picture or slice. I-pictures (slices) are those pictures in which all the MBs are intra-coded. MBs may also be coded with predictions from previous pictures. Pictures (slices), comprising of these types of MBs, are called P-pictures (P-slices). Finally, MBs may be bi- directionally predictively coded from previous and/or future pictures and pictures (slices). Pictures comprising of these types of MBs, are called B-pictures (B-slices). As B-pictures are not used in the prediction loop of the encoder, their loss at the receiver does not noticeably degrade the picture quality. If required, their transmission may be foregone and the bandwidth utilized for a different purpose.

Fig. 1 shows a GoP made up of I-, P- and B-pictures. Normally, a GOP consists of 12 or 15 pictures, taking up about 0.5 s at a frame rate of 25 Hz (frame/s) or 30 Hz respectively. In an H.264/AVC codec, these pictures are specified in the Video Coding Layer (VCL) of the codec.

An H.264/AVC codec conceptually separates the VCL [30] from the Network Abstraction Layer (NAL). This is because the VCL specifies the core compression features, while the NAL supports delivery over various types of network. This network-friendly feature of the standard facilitates easier packetization and improved video delivery. In addition, to adapt H.264/AVC to video applications subject to bit errors and packet losses, a number of error-resilience techniques are provided in the standard. In a communication channel, the QoS is affected by the two parameters of bandwidth and the probability of error.

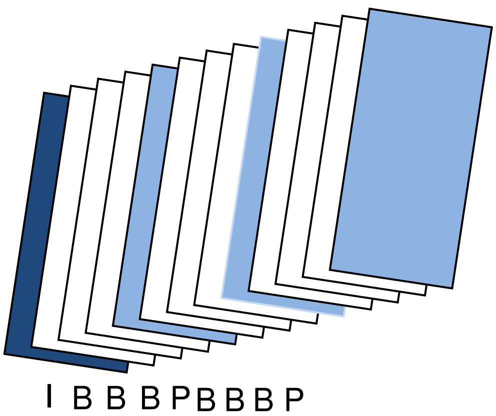


Fig. 1. A group of pictures in H.264/AVC

Therefore, as well as video compression efficiency, which is provided for through the VCL layer, adaptation to communication channels should be carefully considered.

2.2. Network Abstraction Layer

The NAL facilitates the delivery of the H.264/AVC VCL data to the underlying transport layers such as RTP/IP, H.32X and MPEG-2 transportation systems [9]. Each NAL unit (NALU) can be considered as a packet that contains an integer number of bytes including a header and a payload. The header specifies the NALU type and the payload contains the related data. Table 1 is a summarized list of different NALU types. NALUs 1 to 5 contain different VCL data that will be described later. NALUs 6 to 12 are non-VCL units containing additional information such as parameter sets and supplemental information. Parameter sets are header data that remain unchanged over a number of NALUs. Hence, they are transmitted just once. Supplementary information consists of timing and other addressing data that enhances the ability of the decoder to decode but is not essential in decoding the pictures. NALUs 12 to 23 are reserved for future use of H.264/AVC extensions and the types 24 to 31 are unspecified.

In the H.264/AVC codec standard, each frame can be divided into several slices. Each of these contains a flexible number of MBs. Variable Length Coding (VLC), that is entropy coding of the compressed data, takes place as the final stage of a hybrid codec. In H.264/AVC, arithmetic coding replaced the other forms of entropy coding that were used in earlier codecs. In each slice, the arithmetic coder is aligned and its predictions are reset.

Table 1. NAL Unit Types

NAL unit type	*Class*	**Content of NAL unit**
0	-	Unspecified
1	VCL	Coded slice
2	VCL	Coded slice partition A
3	VCL	Coded slice partition B
4	VCL	Coded slice partition C
5	VCL	Coded slice of an IDR picture
6-12	Non-VCL	Suppl. info., Parameter sets, etc.
13-23	-	Reserved
24-31	-	Unspecified

Hence, every slice in the frame is independently decodable. Therefore, slices can be considered as resynchronization points that prevent error propagation to the entire picture. Each slice is placed within a separate NALU (see Table 1). The slices of an Instantaneous Decoder Refresh- (IDR-)[1] or I-picture (i.e. a picture with all intra slices) are located in type 5 NALUs. Those belonging to a non-IDR or I-picture (P- or B-pictures) are placed in NALUs of type 1. However, when DP mode is activated, they are placed in types 2 to 4, as now explained.

In type 1 and type 5 NALs, MB addresses, MVs and the transform coefficients of the blocks, are packed into the packet in the order they are generated by the encoder. In Type 5, all parts of the compressed bitstream are equally important (though for IDR pictures there are no MVs). In type 1, the MB addresses and MVs are much more important than the Discrete Cosine Transform (DCT) coefficients. Notice that in H.264/AVC the DCT is integer-valued to guard against computer arithmetic errors. Symbols appearing earlier in the bitstream suffer less from errors than those which come later[2]. Therefore, in the event of errors in this type of packet, bringing the more important parts of the video data (such as headers and MVs) ahead of the less important data can significantly reduce channel errors. In the standard video codecs [9] prior to H.264/AVC this (bringing the more important data forward) is known as data partitioning.

However, in H.264/AVC, when DP is enabled, every slice is divided into three *separate* partitions and each partition is located in either of type 2 to type-4

[1] An IDR picture is confusedly equivalent to an I-picture in previous standards. An I-picture in H.264/AVC allows predictive references beyond the boundary of a GoP.

[2] Because of the cumulative effect of VLC, symbols nearer the slice synchronization marker suffer less from errors than those that appear later in a bitstream.

NALUs, as listed in Table 1. A NALU of type 2, also known as partition-A, comprises the most important information of the compressed video bit stream of P- and B-pictures, that is the MB addresses, MVs and essential headers. If any MBs in these pictures are intra-coded, their DCT coefficients are packed into the type-3 NALU, also known as partition-B. A type 4 NAL, also known as partition-C, carries the DCT coefficients of the motion-compensated inter-picture coded MBs. It is worth noting that, because in I-slices all MBs are encoded, type-5 NALUs are very long. On the other hand partitions-A and -B of data-partitioned P- and B-slices are smaller but their C-type partition can be relatively long in broadcast-quality video.

DP is a form of source-coded error resilience [26]. Combining, error resilience with error control involves additional data overhead. However, Fig. 2 shows that of four common error resilience tools in H.264/AVC, DP has the least overhead. In Fig. 2, the horizontal axis represents the mean bitstream rate arrived at by setting the quantization parameter (QP) to the given value, while the vertical axis represents the mean overhead rate with that QP. As the quality decreases (higher QP) the advantage of DP increases as the relative overhead of all schemes increases. Tests of the 'Akiyo', 'Coastguard', and 'Mobile' sequences show that the overhead is not strongly dependent on source-coding complexity, with the size of overhead ordering between the schemes preserved. The relative mean sizes (across all frames in the sequence) of the data partitions for a sequence with high spatial coding complexity, 'Paris', and one with high temporal coding complexity, 'Stefan', were examined. The results for these sequences are reported in Table 2 according to video quality given by the QP setting. Both sequences were encoded at Common Intermediate Format (CIF) (352 × 288 pixel/frame), with a Group of Picture (GoP) structure of IPPP….. at 30 Hz (frames/s). Experiments not shown indicate that including B-pictures, with a GoP structure of IPBP (sending order) and

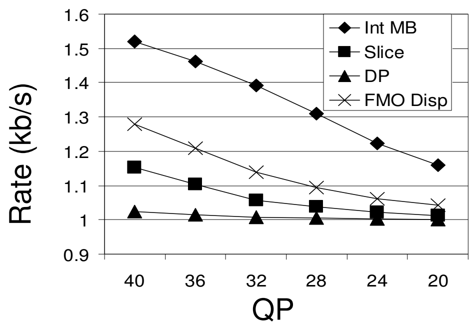


Fig. 2. QCIF 'Foreman' rates according to QP (horizontal axis) plotted against overhead rate (vertical axis) arising from H.264 error resilience tool: Int MB = Intra-coded macroblock refresh, FMO Disp = Flexible Macroblock Ordering with checkerboard (two slices), DP, Slice = slice structuring with 3 slices per frame

Table 2. Relative sizes of partitions A, B, and C for video sequences Paris and Stefan according to video quality

QP	*Paris*			*Stefan*		
	A	B	C	A	B	C
20	11%	9%	80%	5%	5%	90%
30	33%	11%	56%	36%	9%	55%
40	66%	12%	22%	62%	10%	28%

intra-refresh rate of 15, produced similar results.

The relatively small size of partitions-A and -B is a potential advantage at lower QPs but this comes at a cost of a high bitrate. Conversely, at the low quality end of the QP range, (say) QP = 40, if no protection is given to partition-A NALUs, then they become relatively vulnerable to packet loss by virtue of their relatively increased length. However, the scheme described in this paper is most appropriate for higher quality video (here QP = 20, 30), as it assumes a relatively larger partition-C size.

In order to decode partition-B and -C, the decoder must know the location from which each MB was predicted, which implies that partitions B and C cannot be reconstructed if partition-A is lost. Though partition-A is independent of partitions-B and -C, Constrained Intra Prediction (CIP) should be set [6] if partition-B is to be made independent of partition-C. By setting this option, partition-B MBs are no longer predicted from neighboring inter-coded MBs, the prediction residuals of which reside in partition-C. Unfortunately, this option increases the size of packets and reduces compression efficiency, which is why it was not set in the tests in this paper.

There is another dependency [6] arising from Context-Adaptive VLC (CAVLC) entropy coding, because the number of non-zero coefficients in one MB are predicted from the number in a neighboring MB. By design, setting CIP also results in setting the number of non-zero coefficients in data-partitioned inter-coded MBs to zero when CAVLC is in operation to code intra-coded MBs. Thus, partition-B can be made independent of partition-C. It is not possible to employ the alternative Context Adaptive Binary Arithmetic Coding (CABAC), as this option is not supported in the Extended profile of H.264/AVC, though this is the only profile in which data-partitioning is supported. As CAVLC still predicts from intra-coded MBs, when coding partition-C's inter-coded MBs, partition-C cannot normally be made independent of partition-B.

2.2 Unequal Error Protection

The scheme introduced in Section 1 is a form of unequal error protection (UEP), which in general applies to data that can be arranged in a nested set of priorities. Thus, if the highest priority data are not received then lower-priority data are no longer useful, as occurs in Priority-Encoding Transmission (PET) schemes [3]. A variation of UEP in [21] was adapted to rateless codes for fixed-rate transmission. In these circumstances, Unequal Recovery Time (URT) equates to UEP with rateless codes, because of the iterative decoding property possessed by URT. Similarly, the decoding probability is in general also variable under rateless coding across the received data. This is because, unlike traditional codes such as RS, decoding of rateless codes is probabilistic.

In [28], packets were protected by a rateless code according to their picture type, whether I-, P-, or B-picture (refer to Section 2.1). A similar scheme can be found in [5]. In [28], a Raptor code [24] (as further described in Section 2.4) was specified. This scheme directly used rateless coding rather than a window-growth code. Consequently, the protection rates for MPEG-1, -2 video were worked out in advance. That scheme [28] is less flexible than the UEP of data-partitioned packets, as it is most effective when B-pictures are available. In comparison, DP can work if B-pictures are present or not. (Omitting B-pictures reduces decoder complexity on a mobile device.) However, an important contribution of [5] is the concept of probabilistic calculation of the UEP protection and overhead levels for a given GoP configuration, according to the assumed error rate. We now describe rateless codes, which are the basis of the proposed UEP scheme.

2.3 Rateless codes

Rateless coding is ideally suited to a binary erasure channel in which either the error-correcting code works or the channel decoder fails and reports that it has failed. In erasure coding, all is not lost, as flawed data symbols may be reconstructed from a set of successfully received symbols (if sufficient of these symbols are successfully received). A fixed-rate (n, k) RS erasure code over an alphabet of size $q = 2^L$ has the property that if *any* k out of the n symbols transmitted are received successfully then the original k symbols can be decoded. However, in practice not only must n, k, and q be small but also the computational complexity of the decoder is of order $n(n - k) \log_2 n$. The erasure rate must also be estimated in advance.

The class of Fountain codes [16] allows a continuous stream of additional symbols to be generated in the event that the original symbols cannot be decoded. It is the ability to easily generate new symbols that makes Fountain codes rateless. Decoding will succeed with small probability of failure if any of $k\ (1 + \varepsilon)$ symbols are successfully received. In its simplest form, the symbols are combined in an exclusive OR (XOR) operation according to the order specified by a random, low density generator matrix. The probability of decoder failure is $\partial = 2^{-k\varepsilon}$, which for large k approaches the Shannon limit. The random sequence must be known to the receiver but this is easily achieved through knowledge of the sequence seed.

Luby transform (LT) codes [14] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure. The LT's 'belief propagation' decoding relies on the column entries of the generator matrix being selected from a robust Soliton distribution. In the LT generator matrix case, the expected number of degree-one combinations (no XORing of symbols) is $S = c\ \log_e(k/\partial)\sqrt{k}$, for small constant c (see equation (3) later on) Setting $\varepsilon = 2.\log_e(S/\partial).S$ ensures that by sending $k(1 + \varepsilon)$ symbols these symbols are decoded with probability $(1 - \partial)$ and decoding complexity of order $k \log_e k$.

Encoding of the LT in the form used in this paper is accomplished as follows. Choose d_i randomly from some distribution of degrees, where $\rho_{di} = Pr[\text{degree } d_i]$, Pr is the probability. Choose d_i random information symbols R_i among the k information symbols. These R_i symbols are then XORed together to produce a new composite symbol, which forms one symbol of the transmitted packet. Thus, if the symbols are bytes then all of the R_i byte's bits are XORed in their turn with all of the bits of the other randomly selected bytes. It is not necessary to specify the random degree or the random symbols chosen if it is assumed that the (pseudo-)random number generators of sender and receiver are synchronized, as mentioned above.

Symbols are processed at the decoder as follows. If a symbol arrives with degree greater than one it is buffered. If a clean symbol arrives with degree one then it is XORed with all symbols in which it was used in the encoding process. This reduces the degree of each of the symbols to which the degree-one symbol is applied. When a degree-two symbol is eventually reduced to degree-one, it too can be used in the decoding process. Notice again that a degree-one symbol is a symbol for which no XORing has taken place. Notice also that for packet erasure channels a clean degree-one symbol (a packet) is easily established as such because of physical layer checks. In this paper, as sub-packet symbols, namely bytes, are used, then a Cyclic Redundancy Check (CRC) retrospectively determines whether all bytes in a packet have been reconstructed (assuming the CRC is specially protected). It is also assumed that data symbols are checked for successful receipt at the wireless physical layer, before passing them up through the layers of the protocol stack.

The logical degree distribution to use (the ideal Soliton distribution) [16] is given by:

$$\rho(1) = 1/n \qquad (1)$$

$$\rho(d) = \frac{1}{d(d-1)} \qquad d = \{2, 3, \ldots, k\} \qquad (2)$$

where k is the number of source symbols. In practice, the robust Soliton distribution [14] is employed as this produces degree-one symbols at a more convenient rate for decoding. It also avoids isolated symbols that are not used elsewhere. Two tuneable parameters [16] c and δ are used to control the expected number of useable degree one symbols:

$$S = c \ln\left(\frac{k}{\delta}\right)\sqrt{k} \qquad (3)$$

where c is a constant close to 1 and δ is a bound on the probability that decoding fails to complete. Then define

$$\tau(d) = \begin{cases} \frac{S}{k}\frac{1}{d} & \text{for } d = 1, 2, \ldots (k/S)\text{-}1 \\ \frac{S}{k}\ln\left(\frac{S}{\delta}\right) & \text{for } d = k/S \\ 0 & \text{for } d > k/S \end{cases} \qquad (4)$$

as an auxiliary positive-valued function to give the robust Soliton distribution:

$$\mu(d) = \frac{\rho(d)+\tau(d)}{Z} \quad (5)$$

where Z normalizes the probability distribution to unity and is given by $z = \sum_d (\rho(d) + \tau(d))$.

The essential differences between Fountain erasure codes and RS erasure codes are that: (a) Fountain codes in general (not Raptor codes [24]) are not systematic; and that (b) even if there were no channel errors there is a very small probability that the decoding will fail. In compensation, they are completely flexible, have linear decode computational complexity, and generally their overhead is considerably reduced compared to fixed erasure codes. Apart from the startling reduction in computational complexity, a Raptor code has the maximum distance separable property. That is the source packets can be reconstructed with high probability from any set of k or just slightly more than k received symbols. A further advantage of Raptor coding is that it does not share the high error floors on a binary erasure channel [17] of prior rateless codes.

Furthermore, if the packets are pre-encoded with an outer code, a weakened LT transform (one with low average degree) can then be applied to the symbols and their redundant symbols. The advantage of this Raptor code [24] is a decoding complexity that is linear in k. A systematic Raptor code is arrived at by first applying the inverse of the inner code (which should be invertible) to the first k symbols before the outer pre-coding step.

In the multimedia broadcast multicast system (MBMS) [1], Raptor coding at the application layer was introduced by 3GPP for video streaming. However, MBMS differs from the use of rateless coding in our paper because (a) it is obviously for multicast not for unicast streaming, and (b) there is no feedback, because rateless coding is employed for its excellent coding properties rather than because it is rateless. Window-growth codes [29], a further scalable extension of rateless codes, are further described in Section 3.

Window growth codes allow the protection of prioritized data to be incrementally scaled, which is convenient for layered video in general and data-partitioned video in particular. In Section 4, the symbol type is set to a byte but other units are possible, though care must be taken to minimize latency.

3 Window growth codes

This Section describes how window growth codes can be applied to the protection of data-partitioned video streams. Priority data in partitions-A and -B are protected by two sets of overlapping redundant information (which can be of varying extent), while the less important partition-C is protected less.

For I-pictures/slices, a type 5 NAL can be Raptor-coded [24] with redundant information D, as shown in Fig. 3a. The total P-frame allowance was split between a protection group formed by A, B, and C NALUs (redundant symbols marked D in Fig. 3b) and a protection group formed by A and B NALUs (redundant symbols marked E in Fig. 1). In the event of a failure to decode partitions-A and -B, additional redundant data labelled D in Fig. 3b can be applied. As partition-C is less important, if it cannot be decoded using redundant information D, its decoding can be postponed until sufficient additional redundant data arrives.

Our proposed scheme for safe delivery of video stream is as follows. For every k data symbols (type5, A+B, or A+B+C), the Raptor coder generates a rateless redundant data of r symbols. These data can be partitioned into blocks of symbols, as in theory r can be infinitely long to ensure all of the k data symbols can be safely decoded. For transmission purposes, each packet comprises K blocks of data, and the first Y blocks of their redundant data r are sent at the position of Y in Fig 3c.

Assume successfully-received data are passed to the application layer from the physical layer. The packet also includes a

CRC calculated from the *K* blocks. The CRC can be strongly protected by duplication or triplication or by some other means. Recalculation of the CRC at the receiver and comparison with the sent CRC indicates whether the data decode was successful. In event of error, the successfully received transmitted data are stored and in the following packet additional redundant blocks of *r*, identified by X in Fig. 3c are sent. Successfully received additional redundant blocks will help to rectify the failed decoding operation. If the decoder still is not able to decode, more redundant blocks in the following packets will be sent. The process is continued, until each block is safely decoded.

Of course, for a delay-sensitive service such as video, transmission of additional redundant blocks cannot go on forever, and there should be a limit. Our proposed scheme is to confine the decoding delay within a certain number of pictures (e.g. 15 pictures, equal to approximately half a second at a frame rate 30 Hz). To limit the number of transmissions of redundant blocks (X) for previous data, the length of these blocks in the following packets can be gradually increased. For I-pictures/slices, as the length of type 5 NALs can be very long, the lengths of redundant code *r* is much longer than those of P- and B-pictures. As shown in Fig. 1, there are several B-pictures/slices after each I-picture, and instead of transmission of B-pictures, one may just send the redundant D blocks of I-pictures. This is because, as previously remarked in Section 2.1, B-pictures can be easily discarded without significantly impairing video quality.

For P- and B-pictures, since the size of partition-A and -B data is relatively small for high quality video, the number of redundant blocks is also small, and these partitions can easily be decoded after the arrival of a few additional packets. For partitions of type C, though such partitions can be long, because their impact on picture quality is small, they can be easily sacrificed in favor of sending additional redundant blocks E to apply to previous A and B blocks. When no B-pictures are used, to avoid over complex decoders, these C type NALUs can also be replaced in order to carry the redundant data of type 5 for IDR-pictures. This procedure will significantly reduce the decoding delay during severe adverse channel conditions.

Notice that currently, the Constrained Baseline and Baseline profiles of H.264/AVC do not support B-pictures, as they are intended respectively for low-delay codecs, which are used in video conferencing and for mobile devices, which have limited processing power. However, the behavior of these profiles can be emulated in the Extended profile, which currently is the only profile that supports DP.

4 Evaluation

In this Section, the scheme is tested for both a wireless channel and a wired ADSL channel. Burst errors are simulated and the resulting video quality is established to demonstrate the potential advantages of the method. In simulations, by way of comparison, UEP with window-growth codes and DP were compared to an equivalent level of equal error protection

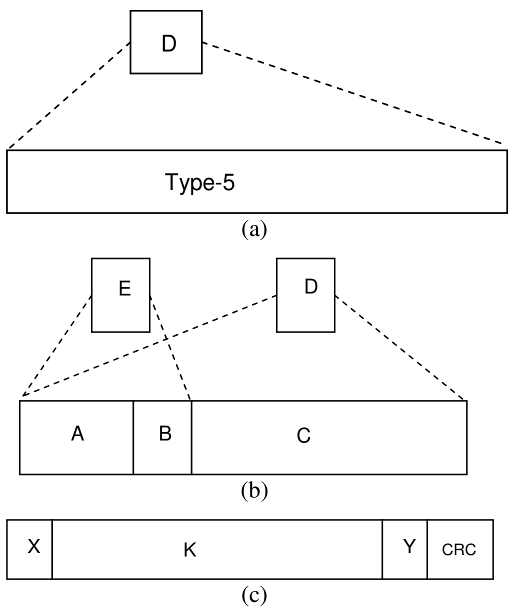


Fig. 3. (a) I-slices and redundant data (b) Redundant codes for data-partitioned video (c) Packetized rateless coded data with CRC

(EEP). EEP was formed by applying rateless codes without DP.To test the performance of the proposed scheme, the standard 'Foreman', and 'Mobile' video sequences, with medium to high motion, were encoded at CIF-30Hz @ 1 Mbps and 4:2:0 chroma sub-sampling. The data that survived simulated transmission were decoded in the Extended Profile with the H.264/AVC JM14.1 decoder software. The GoP size was the normal 15 with IPP… format, that is one I-frame followed by 14 P-pictures. Avoiding B-pictures removes the need for more complex bi-predictive decoding, which can prove a strain for some mobile device processors. With 9 slices per-picture, i.e. two rows of MBs per slice, each P-picture generated 27 NALU-carrying packets of types A, B or C, and each reference IDR-picture resulted in 9 type-5 packets. Calculation of luminance peak signal-to-noise ratio (PSNR) was accomplished through in-house software, as the alternative EvalVid software [10] requires conversion from H.264/AVC to MPEG-4 format, prior to calculation of the PSNR. Each data-point is the average (arithmetic mean) of fifteen runs. For the purposes of these tests, it is assumed that sufficient buffering is present [28] at the receiver to absorb jitter.

IDR-frames were given one and a half as much protection (in terms of redundant rateless bytes) as the total allowance for P-pictures, which was 10% (as in MBMS [1]). Empirical investigations caused us to split equally the total P-picture allowance between a protection group formed by partitions-A, -B, and -C (redundant symbols marked D in Fig. 3) and a protection group formed by partitions-A and -B alone (redundant symbols marked E in Fig. 3). Symbol (byte) erasures were assumed to be detected by the radio receiver. After decoding, it is also assumed that a CRC determines the success of reconstructing with redundant symbols an IDR-picture type-5 NAL. If not, further redundant data blocks are transmitted within the following packets containing P-picture data. This results in a larger window size with a greater probability of successful decoding. If such a window still is not sufficient for decoding the IDR data, partition-C may be seized upon to create more room for additional redundant IDR blocks.

For P-frames, the partitions-A and -B were first decoded with the aid of redundant blocks E. If this decode was successful then partition C is decoded with the aid of redundant blocks D. If decoding of A and B was not successful using E (as judged by a CRC) then decoding of C was postponed until sufficient additional redundant blocks from E are received in the following packets to be able to decode partition-A and-B. Lastly, decoding partition-C is attempted with redundant symbols from D, including additional symbols sent in the following packets.

The number of redundant data transmitted in the following packets for D can be less than that contained in E, as partition C is less important than partitions A and B. However, using the two protection groups also means that two CRCs are required for a P-frame slice (contained in the area marked CRC in Fig. 3c). In the event that, despite all of this, some parts of the pictures are still in error, they can be concealed [9] with the aid of the MVs of the neighboring pixel blocks.

The classic Gilbert-Elliott (G-E) discrete time, two-state, ergodic, hidden-Markov-chain channel model [7] was applied to create erasure bursts, similar to the bursts resulting from slow fading on a wireless channel. If the burst length L is fixed and equal to the average time in a bad state T_B, then the average erasure rate, R, is found as

$$R = \frac{L}{T_G + L} \tag{6}$$

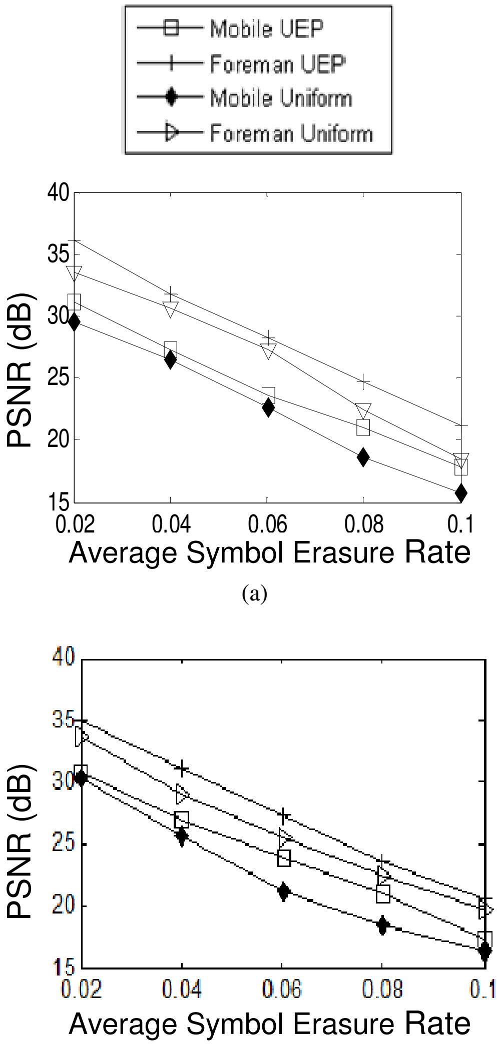
where the average time in the good state with no erasures, $T_{G,}$ is varied according to a desired average erasure rate. The mean times sent in each state are found from (7).

$$T_G = \frac{1}{1 - P_{GG}}, T_B = \frac{1}{1 - P_{BB}} \tag{7}$$

where P_{GG} is the probability that given the current state is good (G), the probability that the next state is also *G* (and similarly for P_{BB} with the other probabilities following from these).

The average symbol (byte) erasure rate was varied, with an average burst size of ten symbols. Finally, to show the relative advantage of rateless code with data partitioning over non-data-partitioned, P-pictures were also uniformly packed into type-1 NALs (no DP). In Fig. 4, the results of rateless coding of data-partitioned video are identified by UEP and those of EEP by 'Uniform'. The Figure shows that for both sequences, UEP gives several dBs improvement in video quality over EEP, and higher coding gains are achieved at higher symbol erasure rates. However, notice that video quality below 25 dB is poor and below 20 dB may well be unwatchable. Comparing Fig.4a with 4b, it can be seen that increasing the average burst size to twenty slightly reduces the video quality. There is also a video content-dependent effect. For example, in Fig. 4a for Foreman at an erasure rate of 0.08 the quality is pushed over the 25 dB threshold. The quality is then within the range tolerated by viewers of mobile TV. Importantly, the improvement is consistent across the range of erasure rates.

In Fig. 5, the UEP scheme results in a greater percentage loss of partition-C carrying packets. Consequently, relatively more protection is afforded to A and B partition packets. In the UEP scheme, because relative to IDR-frame packets less protection is given to A and B partition packets, less IDR-frame packets are lost. From Fig. 5, it should be noted that in the UEP scheme, greater loss of C packets does *not* result in more partition-A and -B carrying packets, a gain from the UEP scheme. The UEP scheme was also applied to a Repetitive Electrical Impulse



(a)

(b)

Fig. 4 Effect of UEP compared to uniform protection (EEP) for increasing symbol erasure rates with an average error burst size of (a) ten (b) twenty symbols (bytes) in a G-E model channel (with a common key for (a) and (b)

Noise (REIN) model channel. For an ADSL channel, REIN was modelled with fixed-length (8 ms) bursts. The bursts were randomly placed to achieve bit error rates between 10^{-7} and 5.10^{-2}. This is the same channel model as used in [15] for ADSL. Within the range of symbol erasure rates of those modeled for a broadband wireless channel, the gain over

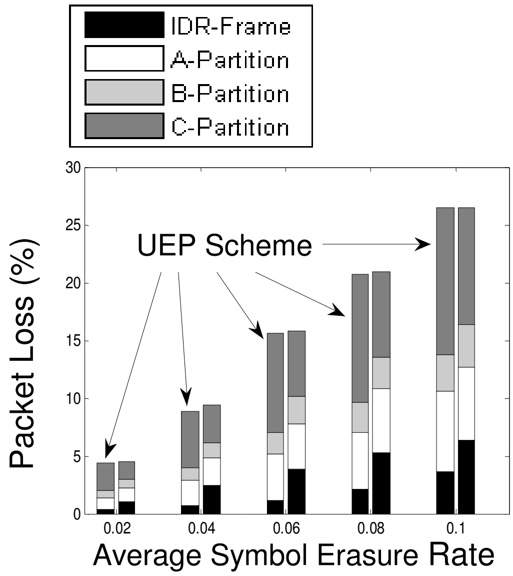


Fig. 5. Packet loss distribution for Mobile clip with average burst length of twenty symbols with UEP to the left and uniform protection to the right

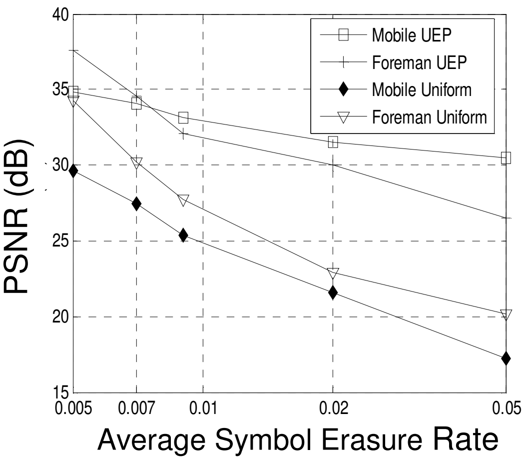


Fig. 6. Packet loss distribution for Mobile clip with average burst length of twenty symbols with UEP to the left and uniform protection to the right

EEP, Fig. 6, was around 10 dB rather than 5 dB, a considerable advantage from applying the proposed scheme to an ADSL channel.

5 Conclusion

The migration to networked video systems poses a problem at the access network, because, as broadband wireless is increasingly employed, error bursts are a threat to a fragile compressed video stream. This paper introduced a protection scheme for the different H.264/AVC NALU types, with particular attention given to the data-partitioned modes. Through gradated protection of more important partitions, a form of layered coding for a unicast video stream results. This is implemented by means of window-growth codes for rateless channel coding. A key feature is the ability to discard less important data, such as residual DCT coefficients of P-pictures, in favor of additional redundant data. The scheme is general, though we have chosen to demonstrate the potential gains at the byte level. These gains consistently raise the quality of delivered video by several dB (according to PSNR), an effect that will definitely be visible to the viewer. To further confirm the value of the proposed scheme, further experimental investigations and analysis will be required, as current investigations have shown the feasibility of the scheme but still need to show the robustness of the proposal.

Acknowledgments The authors gratefully thank Rouzbeh Razavi and Muhammad Altaf for conducting simulations in support of the findings in this paper.

References

1. Afzal J, Stockhammer T, Gasiba T, Xu W (2006) Video streaming over MBMS: a system design approach. J. of Multimedia 1(5): 25–35
2. Ahmad S, Hamzaoui R, Al-Akaidi M (2007) Robust live unicast video streaming with rateless codes. In proceedings of the 16th Int. Packet Video Workshop, Lausanne, Switzerland, pp. 78–84
3. Albanese A, Blömer J, Edmonds J, Luby M, Sudan M (1996) Priority encoding transmission. IEEE Trans. Info. Theory 42(6): 1737–1744
4. ANSI T1.413-1998 (1998) Network and Customer Installation Interfaces – Asymmetric Digital Subscriber Line (ADSL) Metallic Interface. American National Standards Institute
5. Bouabdullah A, Lacan J (2006) Dependency aware unequal erasure protection codes. J. of Ziezhang Univ. 7: 27–33.
6. Dhondt, Y, Mys, S, Vermeirsch, K, Van de Walle, R (2007) Constrained inter prediction: Removing dependencies

between different data partitions. Proceedings of Advanced Concepts for Intelligent Visual Systems, pp. 720–731

7. Fathi H, Chakroborty SS, Prasad R (2008) Models for performance analysis in wireless networks. Voice over IP in Wireless Heterogeneous Networks Signalling, Mobility, and Security, Springer Verlag, pp. 49–57
8. Ghanbari M (1997) Two-layer coding of video signals for VBR networks. IEEE J. on Sel. Areas in Commun. 7(5): 771–781
9. Ghanbari M (2003) Standard video codecs: Image compression to advanced video coding. IET Publications, London, UK
10. Klaue J, Rathke B, Wolisz A (2003) EvalVid - A framework for video transmission and quality evaluation. Proceedings of Int. Conf. on Modeling Techniques and Tools for Computer Performance, pp. 255–272
11. Kumar, A (2007) Mobile TV: DVB-H, DMB, 3G systems and Rich Media applications. Amsterdam: Focal Press
12. Lettieri P, Fragouli C, Srivastava MB (1997) Low power error control for wireless links. Proceedings of the 3rd Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, pp. 139–150
13. Liang, YJ, Apostolopoulos JG, Girod, B (2008) Analysis of packet loss for compressed video: Effect of burst losses and correlation between error frames. IEEE Trans. Circuits Syst. Video Technol. 18(7): 861–874
14. Luby M (2002) LT codes. In proceedings of the 34rd Ann. IEEE Symp. on Foundations of Computer Science, Vancouver, Canada, pp. 271–280
15. Luby M, Stockhammer T, Watson M (2008) Application layer FEC in IPTV systems. IEEE Commun. Mag. 46(5): 94–101
16. MacKay DJC (2005) Fountain codes. IEE Proceedings: Commun. 152(6): 1062–1068
17. Palanki R, Yedidai J (2004) Rateless codes on noisy channels. In proceedings of Int. Symp. Inform. Theory
18. Park, S, Jeong, S-H (2009) Mobile IPTV: Approaches, challenges, standards and QoS support. IEEE Internet Computing, 13(3): 23–31
19. Peng, C, Tan, Y, Xiong, N, Yang, D, Zhang, H, Chao, H-C, et al. (2009) Adaptive video-on-demand broadcasting in ubiquitous computing environment. Springer Journal of Personal and Ubiquitous Computing, 13(7): 479–488.
20. Rahnavard N, Vellambi BN, Fekri F (2007) Efficient broadcasting via rateless coding in multihop wireless networks with local information. In proceedings of the Int. Wireless Communs. and Mobile Computing Conf., Honolulu, Hawaii, pp. 85–95
21. Rahnavard N, Vellambi BN, Fekri F (2007) Rateless codes with unequal error protection property. IEEE Trans. Info. Theory 53(4): 1521-1532
22. Reponen, E, Huuskonen, Mihalic, K (2008) Primary and secondary context in mobile video communication. Springer Journal of Personal and Ubiquitous Computing, 12(4): 281-288
23. Robinson, S, Eslambolchilar, P, Jones M (2010) Exploring casual point-and-tilt interactions for mobile geo-blogging. Springer Journal of Personal and Ubiquitous Computing, 14(4): 363-379
24. Shokorallahi A (2006) Raptor codes. IEEE Trans. Info. Theory 52(6):2551–2567
25. Stockhammer T, Bystrom M (2004) H.264/AVC data partitioning for mobile video communication. Proceedings of the Int. Conf. on Image Processing, Singapore, pp. 545–548
26. Stockhammer T, Zia W (2007) Error-resilient coding and decoding strategies for video communication. In: P. A. Chou, PA, van der Schaar, M (Eds), Multimedia in IP and Wireless Networks, Academic Press, Burlington, MA, pp. 13–58.
27. Sullivan GJ, Wiegand T (2004) Video compression—From concepts to the H.264/AVC standard. Proc. IEEE Dec. 93(1): 18-31.
28. Talari A, Rahnavard N (2009) Unequal error protection rateless coding for efficient MPEG video transmission. In proceedings of Military Communs. Conf., 7 pages
29. Vukobratovic D, Stankovic V, Sejdinovic D, Stankovic L, Ziong Z (2008) Expanding window Fountain codes for scalable video multicast. In proceedings of IEEE Int. Conf. on Multimedia and Expo, Hannover, Germany, pp. 77–80
30. Wenger S (2003) H.264/AVC over IP. IEEE Trans. Circuits Syst. Video Technol. 13:645–656
31. Zorzi M, Rao RR (1997) Error control and energy consumption in communications for nomadic computing. IEEE Trans. Computers 46(2):279–289