Window-growth Rateless Coding for Data-partitioned Video in Wireless Environments

Hannadi Sammek
Communication Department
University of Aleppo
Aleppo, Syria
dr_hanadii@yahoo.com

Martin Fleury and Mohammed Ghanbari
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
{fleum, ghan}@essex.ac.uk

Abstract—**This paper maps channel codes at the application layer to data-partitioned video so that more important compressed data are protected in a closed-loop communication scheme. In particular, window-growth rateless codes are a priority-scalable form of Forward Error Correction that also provide incremental protection to streamed video. The paper introduces a detailed scheme for achieving this according to the H.264/AVC codec's picture types and structures. Consideration is given to the differing data-partitioning modes to establish the feasibility of the scheme for error-prone channels, with a demonstration for a wireless access channel that shows the utility of the scheme, which achieves several dB improvement in video quality (PSNR) through unequal protection compared to equal error protection.**

***Keywords*—data-partitioned video; FEC, rateless channel coding; window growth code**

I. INTRODUCTION

Transmission of video over error-prone channels demands better protection of the compressed bitstream to ensure an acceptable quality of service. Many error-prone channels use Automatic Repeat Request (ARQ), for retransmission of corrupted data. Others may first probe the channel condition to adjust the level of retransmission for better delivery of video [1]. The research in [2] proposed a scheme of error control for wireless channels which varied according to the channel conditions and to the relative energy budget for Reed-Solomon (RS) coding and selective repeat ARQ. However, the volatility of such channels may make estimates unreliable, and, hence, ARQ may not perform well. In addition, retransmission incurs delay that may not be suitable for some streaming applications. However, it is actually possible to protect video against errors without retransmission if layered coding [3] 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 less important lower layers. One form of layered coding is through data-partitioning (DP) [4]. In DP, the compressed video stream is partitioned according to the data priority in terms of the ability to reconstruct 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, as decoding is still possible if essential data such as motion vectors (MVs) arrive intact. In this respect, the family of rateless or Fountain codes [5] appears to be an attractive option for protection of video against channel errors. In this coding method, a varying degree of redundancy is incrementally added to a group of symbols, to ensure that the symbols can be decoded under any 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 channel severity and after a feedback request symbols are gradually transmitted to the receiver until the delay limit is exhausted. Consequently, rateless codes are now attracting applications in video streaming. For example, in [6] rateless coding was applied to packets in unicast video streaming over the Internet and in [7] 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 [4] that can provide good video quality at a small decoding delay. Window-growth codes [8] are an extension of rateless codes which allow the degree of protection to be incrementally scaled. As such they allow prioritized protection of the more important DP partitions. This paper demonstrates an innovatory form of window-growth rateless coding for DP that is better able to reduce transmission delay with reduced redundant overhead. As a demonstration of this property, initial experiments have been applied to a wireless channel subject to burst errors, as such errors frequently occur due to slow and fast fading. However, the main contribution of the current paper is an analysis of how precisely the scheme can be applied to DP in the state-of-the-art H.264/ AVC (Advanced Video Coding) codec [9], as there is a complex arrangement for DP in this codec according to picture type, DP mode, and H.264/AVC profile.

The rest of this paper is organized as follows. Section II comprises background material on rateless codes in general and the organization and structure of the H.264/AVC codec's compressed output. Section III shows how window-growth codes can be applied to a data-partitioned, compressed video bitstream, detailing the scheme for all data partitioned types, including the way retransmission of redundant data takes advantage of non-essential data in the bitstream. Section IV is a simulation to demonstrate the effect of the scheme, while Section V draws some conclusions.

II. BACKGROUND

A. *Unequal error protection*

The scheme introduced in Section I is a form of unequal error protection (UEP), which in general apply 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 [10]. A variation of UEP in [11] was adapted to rateless codes for fixed rate transmission. In these circumstances, Unequal Recovery Time (URT) equates to UEP with rateless codes because of iterative decoding possess URT. Similarly, the decoding probability is in general also variable under rateless coding across the received data (as unlike traditional code, decoding of rateless codes is probabilistic).

In [12], packets were protected by a rateless code according to their picture type, whether I-, P-, or B-picture (refer to Section II.B). A similar scheme can be found in [13]. In [12], a Raptor code [14] (as further described in Section II.B) was specified. This scheme directly used rateless coding rather than a window-growth code and the protection rates for MPEG-1, -2 video are worked out in advance. The scheme is less flexible than UEP of DP packets as it is most effective when B-pictures are available, whereas DP can work if B-pictures are present or not. (Omitting B-pictures reduces decoder complexity on a mobile device.) However, an important contribution is the concept of probabilistic calculation of the UEP protection and overhead levels for a given Group of Pictures (GoP) configuration according to the assumed error rate.

B. *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 [5] allows a continual stream of additional symbols to be generated in the event that the original symbols could not 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 and in this case, 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 [15] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure, provided that the column entries of the generator matrix are 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. 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 $k \log_e k$.

Notice that essential differences between Fountain erasure codes and RS erasure codes are that: Fountain codes in general (not Raptor codes [14]) are not systematic; and that 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.

Furthermore, if the packets are pre-encoded with an inner code, a weakened LT transform can be applied to the symbols and their redundant symbols. The advantage of this Raptor code [14] is a decoding complexity that is linear in k. A systematic Raptor code is arrived at [14] by first applying the inverse of the inner code to the first k symbols before the outer pre-coding step. In the multimedia broadcast multicast system (MBMS) [16], 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 for multicast not for unicast, and (b) there is no feedback, because rateless coding is employed for its excellent coding properties rather than because it is rateless. Moreover, 3GPP systems do not support a feedback channel. 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.

Window-growth codes [8], a further scalable extension of rateless codes, are introduced in Section III after the essential concepts of data-partitioning have first been analyzed. 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 IV, the symbol type is set to a byte but other units are possible, though care must be taken to minimize latency.

C. *Video coding*

A 16×16 pixel block known as a macroblock (MB) is the smallest coding unit of the standard video codecs [17]. 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 where all the MBs are intra-coded. MBs may also be predictively coded with predictions from previous pictures, and pictures (slices) comprising 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) comprising these types of MBs are called B-pictures (B-slices). It is important to note that, since B-pictures are not used in the prediction loop of the encoder, their loss at the receiver does not noticeably degrade the picture quality and if required, their transmission may be foregone and the bandwidth utilized for a different purpose. (We will use this property in our proposed scheme, as described in Section IV.) Fig. 1 shows a group of pictures (GoP) made 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 the H.264/AVC codec [9], these pictures are specified in the Video Coding Layer (VCL) part of the codec.

The H.264/AVC codec conceptually separates [18] the VCL 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-friendliness feature of the standard facilitates easier packetization and improved video delivery. In addition, to adapt H.264/AVC to applications involving bit errors and packet losses, a number of error-resilience techniques are provided in the standard. In a communication channel, the quality of service is affected by the two parameters of bandwidth and the probability of error. Therefore, as well as video compression efficiency, which is provided for through the VCL layer, adaptation to communication channels should be carefully considered. The concept of the NAL, together with the error resilience features in H.264/AVC, allows communication over a variety of different channels.

D. *Network Abstraction Layer*

The Network Abstraction Layer (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 systems [17].

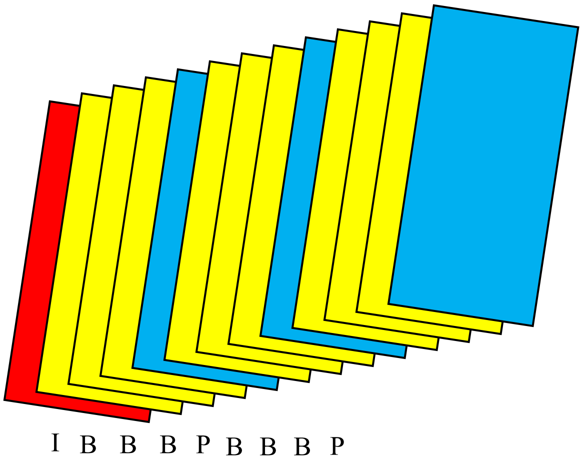


Figure 1. A group of pictures in H.264/AVC

TABLE I. 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

Each NAL unit could be considered as a packet that contains an integer number of bytes including a header and a payload. The header specifies the NAL unit type and the payload contains the related data. Table I is a summarized list of different NAL unit types. NAL units 1 to 5 contain different VCL data that will be described later. NAL units 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 NAL units and, hence, are transmitted just once to prevent repeat transmissions. 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. NAL units 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, each frame can be divided into several slices; each of which contains a flexible number of MBs. Variable Length Coding (VLC) that is entropic coding of the compressed data takes place as the final stage of the hybrid codec. In H.264/AVC arithmetic coding replaced other forms of entropic coding in earlier codecs. In each slice, the arithmetic coder is aligned and its predictions are reset. Hence, every slice in the frame is independently decodable. Therefore, they can be considered as resynchronization points that prevent error propagation to the entire picture. Each slice is placed within a separate NAL unit (see Table I). 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 NAL units, while those belonging to a non-IDR or I-picture (P- or B-pictures) are placed in NAL units of type 1, and in types 2 to 4 when DP mode is active, 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, while in type 1, the MB addresses and MVs are much more important than the Discrete Cosine Transform (DCT) coefficients. In the event of errors in this type of packet, the fact that symbols appearing earlier in the bitstream suffer less from errors than those which come later [2] means that bringing the more important parts of the video data (such as headers and MVs) ahead of the less important data or separating the more important data altogether for better protection against errors can significantly reduce channel errors. In the standard video codecs [17], this is known as data partitioning (DP).

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 NAL units, as listed in Table I. A NAL unit of type 2, also known as partition A, comprises the most important information of the compressed video bit stream of P- and B-pictures, including the MB addresses, MVs and essential headers. If any MBs in these pictures are intra-coded,

---

[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.

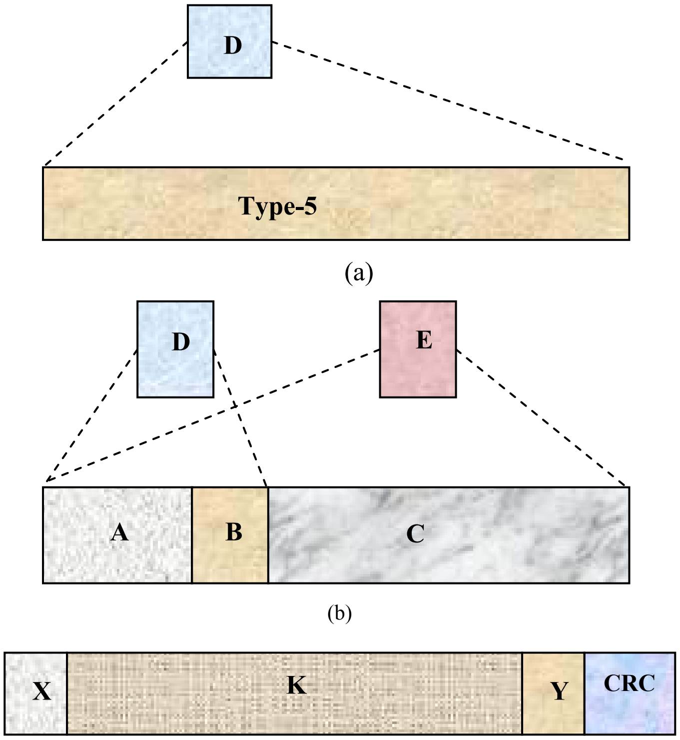
their DCT coefficients are packed into the type-3 NAL unit, also known as partition B. Type 4 NAL, also known as partition C, carries the DCT coefficients of the motion-compensated inter-picture coded MBs. It is worth noting that since in I-slices all MBs are encoded, then type 5 NAL units are very long. On the other hand A and B partitions of data-partitioned P- and B-slices are much smaller but their C-type partition can be very long. In this paper we propose an efficient method for rateless coding of A, B and C type NALUs to make video streaming more efficient.

III. WINDOW GROWTH CODES

For I-slices, a type 5 NAL can be Raptor-coded [14] with redundant information D, as shown in Fig. 2a. For data-partitioned P- and B-pictures, partitions A and B can be Raptor-coded with redundant D, as shown in Fig 2b. For partition C of P- and B-slices, a separate Raptor-code can be applied to the A, B and C partitions with redundant data in E, as shown in the Figure. 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 bits 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 2c. The packet also includes a Cyclic Redundancy Code (CRC) calculated from the K blocks. Recalculation of the CRC at the receiver and comparison with the sent CRC indicates whether the data decode was successful. In case of error, the transmitted data are stored and in the following packet additional redundant blocks of r, identified by X in Fig. 2c are sent. These new redundant blocks will help to decode the failed decoding and if the decoder still is not able to decode, more redundant blocks in the following packets will be sent. The process is continued, until the block is safely decoded.

Of course, for a delay-sensitive service such as video, transmission of additional redundant blocks cannot go on for ever, 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 30 Hz.) To limit the number of transmissions of redundant blocks for previous data (X), the length of these blocks in the following packets can be gradually increased. For I-pictures/slices, where the length of type 5 NALs can be very long, the length of redundant code r is much longer than those of P and B pictures. Fortunately, 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 II.B, B-pictures can be easily discarded without significantly impairing video quality. For P- and B-pictures, since A+B is very small, their number of redundant blocks is also small, and can be easily decoded in a few following packets. For partition type C, though such partitions can be long, since their impact on picture quality is very small, they can be easily sacrificed, for sending the redundant blocks D belonging to previous A+B blocks. For the Baseline profile of H.264, where no B-pictures are used, these C type NALs can also be replaced in order to carry the redundant data of type 5 of I-pictures. This procedure will significantly reduce the decoding delay under severe adverse channel conditions.



(a)

(b)

(c)

Figure 2. (a) I-slices and redundant data (b) Redundant codes for data-partitioned video (c) Packetized rateless coded data with CRC

IV. SIMULATION

In this Section, the scheme is tested for a wireless channel with burst errors and the resulting video quality is reconstructed to demonstrate the advantages of the method. In simulations, by way of comparison, UEP with window-growth codes and DP was compared to an equivalent level of equal error protection (EEP) by rateless codes without DP.

To test the performance of the proposed scheme, the standard 'Foreman', and 'Mobile' video sequences, with medium to high motion, at Common Intermediate Format (CIF)-30Hz @ 1 Mbps, and 4:2:0 sampling were decoded in 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. With 9 slices per-picture, i.e. two rows of MBs per slice, each P-picture generated 27 NALU-bearing packets of type A, B and 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 [19] 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 [12] at the receiver to absorb jitter.

IDR-frames were accorded 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 [16]). 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 E in Fig. 2) and a protection group formed by partitions A and B (redundant symbols marked D in Fig. 2). 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 an IDR-picture type-5 NAL with redundant symbols. If not, they are transmitted in the following packets of P-pictures with a larger window size. If such a window still is not sufficient for decoding IDR data, partition C may be seized upon to create more room for the remaining redundant IDR blocks.

For P-frames, the partitions A and B are first decoded with the aid of redundant blocks D. If this decode is successful then partition C is decoded with the aid of redundant blocks E. If decode of A and B is not successful using D (as judged by a CRC) then decoding of C is postponed until A+B receive sufficient redundant blocks in the following packets, to be decodable. Lastly, decode of partition C is attempted with redundant symbols from E. The number of redundant data transmitted in the following packets for E can be less than D, as partition C is less important than partitions A and B, though the length of E can be larger than D. However this also means that two CRCs are required for a P-frame slice. In the case that, despite all this, still some parts of the pictures are erroneous, they can be concealed [17] with the aid of the MVs of the neighboring pixel blocks.

The classic Gilbert-Elliott (G.-E.) discrete time, two-state ergodic Markov chain channel model [20] 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

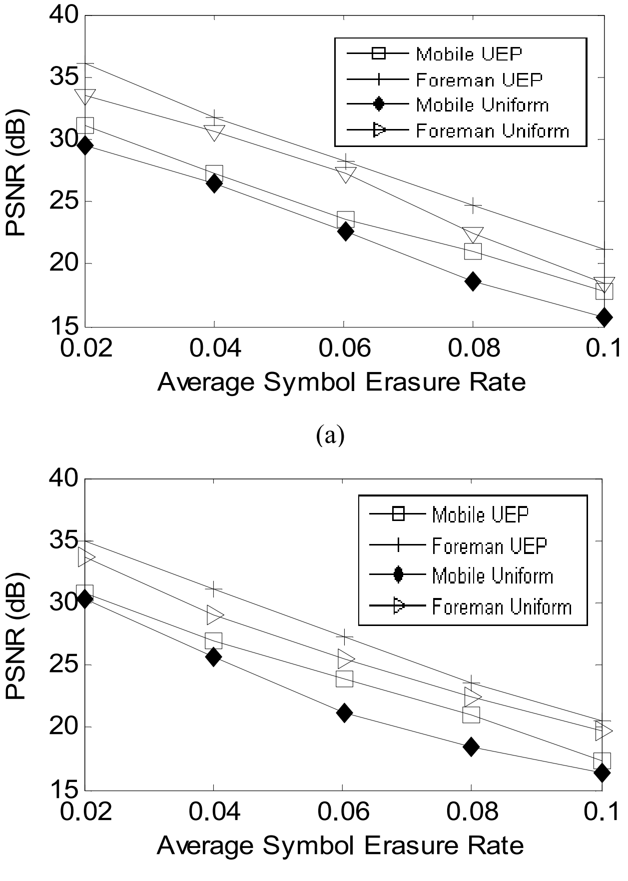
$$R = \frac{L}{T_G + L} \qquad (1)$$

where the average time in the good state with no erasures, T_G, is varied according to a desired average erasure rate. The state transition probabilities are found from (2).

$$T_G = \frac{1}{1 - P_{GG}}, T_B = \frac{1}{1 - P_{BB}} \qquad (2)$$

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-DP, P-pictures were alternatively uniformly packed into type-1 NAL (no data partitioning). In Fig. 3 results of rateless code 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, though video quality below 25 dB is poor and below 20 dB may well be unwatchable. Comparing Fig.3a with 3b, it can



(a)

(b)

Figure 3. 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.

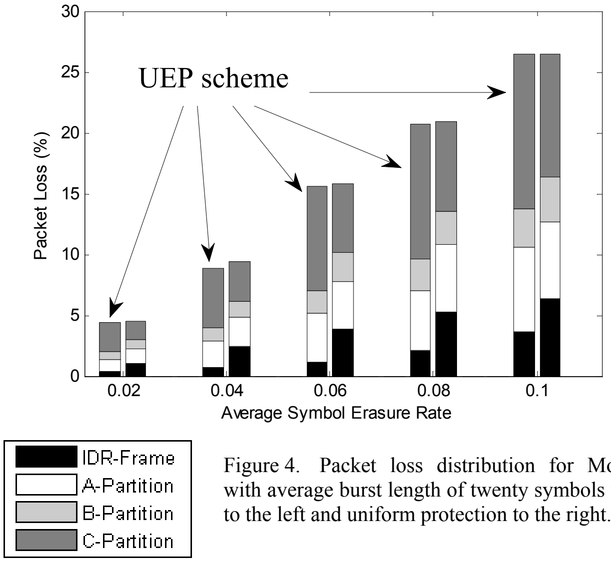


Figure 4. Packet loss distribution for Mobile clip with average burst length of twenty symbols with UEP to the left and uniform protection to the right.

be seen that increasing the average burst size to twenty slightly reduces the video quality.There is a content-dependent effect. For example, in Fig. 3a for Foreman at an erasure rate of 0.08 the quality is pushed over the 25 dB threshold that is within the range tolerated by viewers of mobile TV. Importantly, the improvement is consistent across the range of erasure rates.

In Fig. 4, the UEP scheme results in a greater percentage loss of partition C packets with consequently relatively more protection afforded 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. 4, it should be noted that in the UEP scheme, greater loss of C packets does *not* result in more A and B packets, a gain from the UEP scheme.

V. CONCLUSIONS

The migration to networked IPTV systems poses a problem at the access network, because, when broadband wireless is often employed, error bursts are a problem to a fragile compressed video stream. This paper introduces a protection scheme for the different H.264/AVC NAL unit 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 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 consistently raise the quality of delivered video from by several dB (according to PSNR) an effect that will definitely be visible to the viewer. Further work will involve comprehensive testing of the scheme.

REFERENCES

[1] M. Zorzi and R. R. Rao, “Error control and energy consumption in communications for nomadic computing,” *IEEE Trans. on Computers*, vol. 46, no. 3, pp. 279–289, 1997.

[2] P. Lettieri, C. Fragouli, and M. B. Srivastava, “Low power error control for wireless links,” in *Proc. of the 3rd Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom ’97)*, pp. 139– 150, Sept. 1997.

[3] M. Ghanbari, “Two-layer coding of video signals for VBR networks”, *IEEE J. on Selected Areas in Communications*, Special Issue on Packet Speech and Video, vol. 7, no. 5, pp. 771–781, June 1989.

[4] T. Stockhammer and M. Bystrom, “H.264/AVC data partitioning for mobile video communication,” in *Proc. of Int. Conf. on Image Processing*, pp. 545-548, Singapore, Oct. 2004.

[5] D. J. C. MacKay, “Fountain codes,” *IEE Proceedings: Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[6] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, “Robust live unicast video streaming with rateless codes,” in *Proc. of the 16th Int. Packet Video Workshop*, pp. 78–84, Lausanne, Switzerland, Nov. 2007.

[7] N. Rahnavard, B. N. Vellambi, and F. Fekri, “Efficient broadcasting via rateless coding in multihop wireless networks with local information,” in *Proc. of the Int. Wireless Communs. and Mobile Computing Conf.*, pp. 85–90, Honolulu, Hawaii, Aug. 2007.

[8] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Ziong, “Expanding window Fountain codes for scalable video multicast”, in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, pp. 77-80, Hannover, Germany, April 2008.

[9] G. J. Sullivan, and T. Wiegand, “Video compression -From concepts to the H.264/AVC standard”, *Proc.. of the IEEE*, vol. 93, no. 1, pp. 18-31, Dec. 2004

[10] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, “Priority encoding transmission,” *IEEE Trans. on Informa-tion Theory*, vol. 42, no. 6, pp.1737-1744, 1996.

[11] N. Rahnavard, B.N. Vellambi, and F. Fekri, “Rateless codes with unequal error protection property,” *IEEE Trans. on Information Theory*, vol. 53, no. 4, pp. 1521-1532, 2007.

[12] A. Talari and N. Rahnavard, “Unequal error protection rateless coding for efficient MPEG video transmission,” in *Proc. of Military Communs. Conf.*, 7 pages, 2009.

[13] A. Bouabdullah and J. Lacan, “Dependency aware unequal erasure protection codes,” J. of Ziezhang Univ., vol. 7, pp. 27-33, 2006.

[14] A. Shokorallahi, “Raptor codes”, *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.

[15] M. Luby, “LT codes,” in *Proc. of the 34rd Annual IEEE Symp. on Foundations of Computer Science*, pp. 271–280, Vancouver, Canada, Nov. 2002.

[16] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, “Video streaming over MBMS: a system design approach,” *Journal of Multimedia*, vol. 1, no. 5, pp. 25–35, 2006.

[17] M. Ghanbari, *Standard video codecs: Image compression to advanced video coding*, IET Publications, London, UK, 2003.

[18] S. Wenger, “H.264/AVC over IP,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, , pp. 645-656, July 2003.

[19] J. Klaue, B. Rathke, and A.Wolisz, “EvalVid - A framework for video transmission and quality evaluation,” in *Proc. of Int. Conf. on Modeling Techniques and Tools for Computer Performance*, pp. 255-272, 2003.

[20] H. Fathi, S. S. Chakroborty, and R. Prasad, “Models for performance analysis in wireless networks,” in *Voice over IP in Wireless Heterogeneous Networks Signalling, Mobility, and Security,* publ. Springer, pp. 49-57, 2008.