

Research Article

Protecting H.264/AVC Data-Partitioned Video Streams over Broadband WiMAX

Laith Al-Jobouri, Martin Fleury, and Mohammed Ghanbari

School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

Correspondence should be addressed to Martin Fleury, fleum@essex.ac.uk

Received 16 April 2012; Revised 6 July 2012; Accepted 6 July 2012

Academic Editor: Martin Reisslein

Copyright © 2012 Laith Al-Jobouri 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.

Broadband wireless technology, though aimed at video services, also poses a potential threat to video services, as wireless channels are prone to error bursts. In this paper, an adaptive, application-layer Forward Error Correction (FEC) scheme protects H.264/AVC data-partitioned video. Data partitioning is the division of a compressed video stream into partitions of differing decoding importance. The paper determines whether equal error protection (EEP) through FEC of all partition types or unequal error protection (UEP) of the more important partition type is preferable. The paper finds that, though UEP offers a small reduction in bitrate, if EEP is employed, there are significant gains (several dBs) in video quality. Overhead from using EEP rather than UEP was found to be around 1% of the overall bitrate. Given that data partitioning already reduces errors through packet size reduction and differentiation of coding data, EEP with data partitioning is a practical means of protecting user-based video streaming. The gain from employing EEP is shown to be higher quality video to the user, which will result in a greater take-up of video services. The results have implications for other forms of prioritized video streaming.

1. Introduction

Portable devices are proliferating, as the era of the wired Internet draws to a close and 4G wireless systems, and their successors [1] bring greater bandwidth capacity to access networks. User-based video-streaming applications are anticipated to be a key to the success of broadband wireless access networks such as IEEE 802.16e (mobile WiMAX) [2]. WiMAX itself is proving to be attractive in many areas where existing cell phone coverage is sparse or nonexistent. However, the migration of Internet applications to 4G wireless access presents a problem for video-streaming applications. This is because wireless channels are fundamentally error prone, whereas compression, for most of its coding gain, depends upon predictive coding. Consequently, because of source-coding data dependencies, errors can disrupt a compressed video bitstream, and these errors can subsequently propagate in space and time. In the multimedia research world, unequal error protection (UEP) through

channel coding or forward error correction (FEC) has proved to be a rich area of investigation. Many schemes (some of which are reviewed in Section 2) have been proposed that map differential protection onto prioritized coded video data. However, there are strong signs that, in the commercial world, video service providers, in the interests of video content integrity, have opted for reliable streaming protocols, which simply resend data found to be corrupted or lost. This approach is not possible for all types of service but equal error protection (EEP) up to a sufficient level is possible. At the heart of this paper's investigation is a rather fundamental question, which is whether UEP gains in reducing bitrate are worth the extra complexity involved. One can go further and suggest that EEP will, for a relatively small increase in bitrate, bring significant gains in video quality. It can also avoid computationally intense optimization procedures that may prove unattractive to commercial providers. The current paper demonstrates these ideas in the context of prioritized data-partitioned video streams. As the research community

has naturally investigated UEP procedures, we believe that advocating EEP is a relatively novel approach.

For two-way, interactive applications and user-to-user streaming, the problem of wireless errors cannot be overcome by the currently popular Dynamic Adaptive HTTP Streaming (DASH) [3]. DASH employs reliable TCP transport. However, mobile devices do not have storage capacity for multiple representations of a video stream, as required at DASH servers. For example, in [4] the DASH server storage was found for 90-minute videos encoded at up to 16 bitrates, in steps of 500 kbps starting at 500 kbp. The storage costs were 5 streams at 5 GB, 10 streams at 18 GB, and 16 streams at 46 GB. With current server storage costs [4] as low as 0.125 USD per month per 1 GB, multiple videos can be stored on a server in this way. Unfortunately, even short video clips stored in this way on a mobile device can pose an extra burden on memory capacity, which also has other calls on its capacity. Consequently, the streaming of video sequences with significant source-coding complexity remains particularly at risk, because of increased predictive data dependencies between packets and because of increased packet sizes. Such videos will be temporally or spatially active or a mixture of both.

In previous research by the authors [5], data-partitioned video streaming was employed as a means of separating out the more important source-coded data. In such data-partitioned video, the compressed video bitstream is split into up to three partitions before packetization, according to the importance of the content type to the decoding of the video. In general, smaller, less error-prone, packet sizes result and, for broadcast quality video, the more important data are carried in the smallest packets. In our previous work [5], all such packets were protected against errors with EEP, irrespective of their size. However, it is also possible [6] to apply UEP by duplicating one or more of the higher-priority segments but not duplicating the less important packets. Additionally, it is feasible [7] to protect higher-priority segments through the differential use of scalable channel coding, namely, by means of Raptor rateless coding [8]. However, it is unclear to what extent lower-priority segments can be left unprotected without an adverse effect on video quality or, indeed, whether lower complexity EEP is preferable at a small increase in bitrate. Consequently, the current paper directly compares EEP with UEP by carefully selecting appropriate configurations for data-partitioned video streams.

In work in [9], UEP of data-partitioned video was compared with EEP for single-layer video. Thus, EEP was not applied to data partitioning, as the intention of the work [9] was to show the potential advantage of the limited layering that data-partitioning represents. In [9], UEP was found to provide lower average quality than EEP but it had a greater probability of providing good quality video, despite adverse channel conditions. This leads to the question of why not apply EEP to a data-partitioned video stream.

In [9], differential protection was achieved by selecting from a set of discrete channel coding rates, through punctured convolutional codes. However, in order to determine the protection level, an optimization procedure was

necessary to minimize potential distortion. This procedure depended on the quantization parameter (QP) and the coding rate for each partition. The wireless channel characteristics also had to be known in advance by the encoder. However, leaving aside the computational complexity of the optimization search in [9], there is another key difference between the system of [9] and that of [5] and this paper. In [9] no feedback occurs, so that it is not possible to request additional redundant data. In fact, when using punctured convolutional codes in [9] (rather than the rateless codes used herein), it is not possible to generate additional redundant data. In fact, as discussed in Section 3, rateless channel coding has a number of other advantages over conventional codes, apart from the ability to dynamically generate additional redundant data. We have demonstrated the scheme for WiMAX. The frame structure of WiMAX includes a send and receive subframe, making it convenient to immediately send a single request for additional redundant data. However, for two-way conversational video services such as video-phone, the feedback channel is automatically available anyway.

Data partitioning in this paper can be viewed as a simplified form of SNR or quality layering [10]. Extended quality layering can also be applied to video streaming across WiMAX. In [11], adaptive multicast streaming was proposed using the Scalable Video Coding (SVC) extension for H.264 [12]. Fixed WiMAX channel conditions were monitored in order to vary the bitrate accordingly. Unfortunately, the subsequent decision of the JVT standardization body for H.264/AVC *not* to support fine-grained scalability (FGS) implies that it will be harder to respond to channel volatility in the way proposed in [11]. Other works have also investigated combining scalable video with multiconnections in [13] and in comparison with H.264/AVC in [14]. However, the data dependencies between layers in H.264/SVC medium-grained scalability are a concern. Unlike in FGS, enhancement layer packets may successfully arrive but be unable to be reconstructed if key pictures also fail to arrive. Besides, for commercial one-way streaming, simulcast is now likely to be preferred to H.264/SVC for the reasons outlined in [4]. In [4], it was found that the extra overhead from sending an SVC stream compared to an H.264/AVC stream meant that the cost of bandwidth consumption outweighed the reduced storage cost of SVC once more than 64 sessions had occurred (assuming 16 simulcast streams or 16 video layers per session). In another comparison [15], it was proposed that scalable video with UEP cannot provide any advantage over H.264/AVC with EEP in a wireless environment, due to the overhead of scalable video coding compared to that of single-layer coding.

In an H.264/AVC (Advanced Video Coding) codec, when data partitioning is enabled, every slice is divided into three separate partitions, and each partition is located in either of type-2 to type-4 Network Abstraction Layer Units (NALUs). (A slice is a subdivision of a picture or video frame, and an NALU is output as a virtual packet by an H.264/AVC codec, as part of its network-friendly approach [16].) For simplicity of interpretation just one slice per frame was

employed in the current paper. It is then optionally possible to divide each slice into up to three data partitions. For purely intracoded video frames, I-frames, just two data partitions occur. However, in streaming over wireless it is common to avoid periodic I-frames, as they result in an increased data rate due to the inefficiency of intracoding. Consequently, an IPPPP... frame coding structure (i.e., one I-frame followed by all P-frames) is used with some form of distributed intrarefresh [17]. Then, apart from the first frame, all slices are divided into three.

In such a stream, a packet bearing an NALU of type 2, also known as data-partition-A, contains the most important information, including the Macroblock (MB) types and addresses, motion vectors, and essential header information. If any MBs in these frames are intracoded, their frequency transform coefficients are packed into a type-3 NALU, also known as data-partition-B. Intracoded block patterns (CBPs) are also included, as these specify in compact form which blocks within an MB contain nonzero coefficients. Type-4 NALs, also known as data-partition-C, carry the transform coefficients of the motion-compensated inter-picture coded MBs along with inter-CBPs. These three partitions, types A, B, and C, form segments of the video bitstream. They are subsequently each output as Real-Time Transport Protocol (RTP) packets by the codec in RTP mode, prior to dispatch as Internet Protocol (IP)/User Datagram Protocol (UDP) packets. (It is assumed that header compression over a broadband wireless link will greatly reduce the header overhead [18] from 40 B to one or two B on average.)

Because the evaluation in this current paper uses distributed intrarefresh rather than periodic intracoded frames, delay arising from the sudden dispatch of multiple packets forming I-frames is avoided. As no B-frames are used, the schemes are suitable for the low-complexity processors on mobile devices, though there is an issue over the need for a hardware implementation of data partitioning. Then, by adopting Constant Bit-Rate (CBR) streaming in tests, a comparison between different schemes is fair. In fact, CBR streaming allows commercial providers to plan storage capacity and bandwidth utilization, at a cost in some fluctuations in video quality. From [19], when using data-partitioned video streaming, it is important to set constrained interprediction (CIP), as otherwise partition-B cannot be made completely independent of partition-C. When CIP is set, intraprediction can only be performed by referencing other intracoded MBs. If no suitable MBs are available, then intraprediction is not possible. As CIP prevents predictive reference to inter-coded MBs, the information in partition-C is no longer required, thus allowing partition-B to become independent of partition-C. In the Joint Model (JM) reference software for H.264/AVC, CIP is actually set in the input parameter file. In [20] it is revealed that, even when data partitioning is not in use, setting CIP is effective in combating higher packet loss rates. However, whenever CIP is set, there is a limited loss of compression efficiency, whose loss is quantified in [20]. On the other hand, it is not possible to make partition-C independent of partition-B without breaking the codec's compatibility with

the H.264/AVC standard. Reconstruction of all partitions is dependent on the survival of partition-A, though that partition remains independent of the other partitions.

The remainder of this paper is organized as follows. Section 2 describes physical and software approaches to UEP. Physical (PHY-) layer UEP avoids bitrate overhead but is inflexible compared to software UEP. Section 2 also reviews application-layer EEP in wireless video streaming. Section 3 goes on to consider rateless channel coding, which is employed in adaptive fashion for EEP and UEP alike. Unlike conventional channel coding, in rateless coding, the redundant data to information data ratio can be dynamically scaled, making it suitable for application-layer protection. Then, before a comparative evaluation, Section 4 examines the simulation model and its validity. Section 5 is our comparison of UEP with EEP for data-partitioned video. Section 6 concludes the paper, with some recommendations for future research.

2. Related Research

The idea of UEP for segmented video bitstreams has taken various forms prior to the H.264/AVC codec standard (otherwise known as MPEG-4 part 10). In an MPEG-4 Part 2 codec, partitioning was internal to a packet with just two partitions. The first contained header, motion, and other shape information. The second contained the texture (transform coefficients), with decoder resynchronization headers placed internally at the start of each partition. In [21], PHY-layer FEC was enhanced for a fixed-sized part at the start of each packet. Unfortunately, as the size of the first MPEG-4 partition may vary in size, some motion vectors could receive less protection. Besides, each network traversed by the video stream would need to have special arrangements for this type of traffic. Finally, by placing both partitions in one packet, no account is taken of the risk of decoder desynchronization when packet loss occurs.

To avoid these problems, the authors of [22] proposed that MPEG-4 part 2 internal partitions should be split between packets, forming two different streams. Headers would be needed to allow partitions from the same video frame to be identified. This is what now occurs within an H.264/AVC codec; except three rather than two streams are formed. In [22], UEP was implemented by placing each of the MPEG-4 part 2 streams in different General Packet-Radio Service (GPRS) channels, with different channel coding rates for each stream. However, in our scheme we prefer application-layer protection, in addition to any PHY-layer protection that may be present. This makes a solution more amenable to end-to-end control.

In [23], another approach for broadcast video was taken in which hierarchical modulation favored those H.264/AVC partitions containing more important data for the reconstruction of the video frame. One reason H.264/AVC data partitioning was chosen, rather than other forms of layering, was that it does not significantly increase the bitrate of the composite stream. In fact, this is the same reason that Hierarchical Quadrature Amplitude Modulation (HQAM)

was chosen rather than channel coding: that it does not increase the bitrate. However, in extensions to the scheme, Turbo channel coding was additionally required for poor wireless channel conditions. The proposed scheme [23] was intended to be flexible, altering the QAM symbol constellation according to the desired bitrates.

HQAM is not the only form of PHY-layer prioritization, and in [24] data partitions were mapped onto different antennas in a space-time block coding. Two segments were employed with high-priority bits (those separated more in the coding) for partition-A and low-priority bits for the partitions-B and -C. The prioritization is different from the arrangement in the current paper, because herein partition-A and -B are grouped as a high-priority segment. However, this is explained by the different picture coding structures in each paper, that is, in [24] and the current paper. In the current paper, the use of distributed intrarefresh MBs rather than periodic intracoded pictures (I-pictures) means that it is important to protect partition-B packets, as they contain intracoded transform coefficients.

Software approaches to UEP may combine prioritized channel encoding of video with interleaving across packets. (Interleaving is employed to counter long error bursts during deep wireless channel fades.) In Priority Encoding Transmission (PET) [25], parity symbols of a systematic code are included in successive packets such that high-priority segments can be recovered, even if a large number of packets are erased. On the other hand, lower priority segments will be lost if a few packets amongst the interleaved group are erased. PET is capable of refinement in a rate-distortion manner [26] but, with just three partitions, the relevance of such refinements to the current scheme appears restricted. Besides, a problem with all packet-interleaving methods is the impact of increased latency when the decoder has to wait for all the packets in an interleaved group to arrive before reconstruction can take place.

Turning to EEP, application-layer EEP leads to an increase in overall bitrate. In return, EEP can result in gains in flexibility and in the ability to address the special needs of compressed video arising from the risk of temporal error propagation. Application-layer Raptor code has been applied [27] to a number of error-prone network environments, because of the stringent anticipated requirements for IPTV [28]. In these realizations all packets are protected against erasure, while bit errors are assumed to be protected at the physical layer. The Digital Video Broadcast (DVB) project has specified [29] optional application-layer rateless coding, as has 3rd Generation Partnership Project (3GPP) [30]. However, in these standards the potential for dynamic adaptation of the protection level was not exploited.

3. Rateless Channel Coding

In this paper, rateless coding is employed to protect data-partitioned video. Rateless coding is employed in an adaptive manner [5] by retransmission of additional redundant data, as and when required. However, notice that rateless codes are a probabilistic channel code, in the sense that reconstruction

is not guaranteed. Raptor coding [8], as used herein, is a systematic variety of rateless code that does not share the high error floors of prior rateless codes. It also has $O(n)$ decoder computational complexity. Systematic codes allow packets without any reported errors to be treated separately to those without them. Thus, processing can be sped up by splitting processing into two processing streams if systematic coding is used.

It is the ability to easily generate new symbols that makes rateless codes to be rateless. Decoding will succeed with small probability of decoder failure if any of $k(1 + \epsilon)$ symbols are successfully received, where k is the number of source symbols originally present and ϵ is a low percentage of coding overhead. In its simplest form, the symbols are combined in an exclusive OR (XOR) operation according to the order specified by a randomized, low-density generator matrix, and, in this case, the probability of decoder failure is $\partial = 2^{-k\epsilon}$, which for large k approaches the Shannon limit. The random sequence must be known to the receiver but this is easily achieved through advance knowledge of the sequence seed.

In general, encoding of rateless codes 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 of a given event. Choose d_i random information symbols R_i from amongst 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, all of the R_i byte's bits are XORed with all of the bits of the other randomly selected bytes in turn. 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.

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 decrements the degree of each of the symbols to which the degree-one symbol is applied. When a symbol is eventually reduced to degree one, it too can be used in the decoding process. Notice 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. For byte-erasures, the PHY-layer FEC can be reasonably expected to isolate clean symbols or blocks of clean symbols.

In the decoding process, the robust Soliton distribution [31] is employed as the degree-distribution, as this produces degree-one symbols at a convenient rate for decoding. It also avoids isolated symbols that are not used elsewhere. Two tuneable parameters c and δ serve to form the expected number of useable degree-one symbols. Set

$$S = c \ln\left(\frac{k}{\delta}\right) \sqrt{k}, \quad (1)$$

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

$$\begin{aligned}\tau(d) &= \frac{S}{k} \frac{1}{d} \quad \text{for } d = 1, 2, \dots, \left(\frac{k}{S}\right) - 1 \\ &= \frac{S}{k} \ln\left(\frac{S}{\delta}\right) \quad \text{for } d = \frac{k}{S} \\ &= 0 \quad \text{for } d > \frac{k}{S}\end{aligned}\quad (2)$$

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

$$\mu(d) = \frac{\rho(d) + \tau(d)}{z}, \quad (3)$$

where z normalizes the probability distribution to unity and is given by

$$z = \sum_d (\rho(d) + \tau(d)). \quad (4)$$

4. Simulation Model

4.1. Wireless Configuration. To establish the behavior of rateless coding under WiMAX, the ns-2 simulator was augmented with a module from the Chang Gung University, Taiwan [32] that has proved an effective way of modeling IEEE 802.16e's behavior. Ten runs per data point were averaged (arithmetic mean), and the simulator was first allowed to reach steady state before commencing testing.

In the evaluation, transmission over WiMAX was carefully modeled. The PHY-layer settings selected for WiMAX simulation are given in Table 1. The antenna heights are typical ones taken from the standard [33]. The antenna is modeled for comparison purposes as a half-wavelength dipole, whereas a sectored set of antenna on a mast might be used in practice to achieve directivity and, hence, better performance. The IEEE 802.16 Time Division Duplex (TDD) frame length was set to 5 ms, as only this value is supported in the WiMAX forum simplification of the standard. The data rate results from the use of one of the mandatory coding modes [2, 33] for a TDD downlink/uplink subframe ratio of 3:1. The WiMAX base station (BS) was assigned more bandwidth capacity than the uplink to allow the BS to respond to multiple mobile subscriber stations (MSs). Thus, the parameter settings in Table 1 such as the modulation type and PHY-layer coding rate are required to achieve a data rate of 10.67 Mbps over the downlink. Notice also that there is 1/2 channel coding rate at the PHY-layer of IEEE 802.16e, in addition to the application-layer channel coding that we add. However, as discussed in Section 2, application-layer coding for compressed video stream is frequently used in wireless systems because of the high packet losses and error rates that can occur.

A two-state Gilbert-Elliott channel model [34] simulated the channel model for WiMAX. Though this model does not reproduce the physical characteristics that give rise to noise and interference, it does model the error bursts [35]

TABLE 1: IEEE 802.16e parameter settings.

Parameter	Value
PHY	OFDMA
Frequency band	5 GHz
Bandwidth capacity	10 MHz
Duplexing mode	TDD
DL/UL subframe ratio	3:1
Frame length	5 ms
Max. packet length	1024 B
Raw data rate (downlink)	10.67 Mbps
IFFT size	1024
Modulation	16-QAM 1/2
Guard band ratio	1/16
MS transmit power	245 mW
BS transmit power	20 W
Approx. range to SS	1 km
Antenna type	Omni-directional
Antenna gains	0 dBD
MS antenna height	1.2 m
BS antenna height	30 m

OFDMA: orthogonal frequency division multiple access.

commonly experienced by an application. It is such bursts that are particularly harmful to compressed video data. In the Gilbert-Elliott model, PGG is the probability of remaining in the good state, while PG is the probability of byte error in the good state, which was modelled internally by a Uniform distribution. PBB and PB are the corresponding parameters for the bad state.

4.2. Video Configuration. Two video clips with different source-coding characteristics were employed in the tests in order to judge the dependency of the results upon video source-coding complexity. The first test sequence was *Paris*, which is a studio scene with two upper body images of presenters and moderate motion. The background is of moderate-to-high spatial complexity leading to larger slices. The other test sequence was *Football*, which has rapid movements and consequently has high temporal coding complexity. Both sequences were CBR encoded at Common Intermediate Format (CIF) (352×288 pixel/picture). CIF resolution was used for ready comparison with the prior work of others on video communication with mobile devices.

Clearly if one of the high-definition (HD) resolutions were to be used, as processing within H.264/AVC is on an MB-basis, the number of packets output would normally be scaled up linearly. However, because viewers are more sensitive to visual artefacts at higher resolutions, the frame rate is usually increased from as low as 24 frame/s to as much as 90 frame/s. The fidelity extension to H.264/AVC [36] extended the sample bit depth to ten bits and introduced a new 8×8 transform block size for increased sensitivity to texture detail. An increased frame rate and bit depth

will lead to more than just a linear increase in the number of packets, as would adoption of one of the new chroma formats [36]. This increase requires redimensioning of the buffer at the mobile device to avoid excess packet loss but to not result in an increase in latency at the same time. As streaming rates of greater than 2.5 Mbps for 1280×720 pixels/frame progressively scanned (720p) HD video [37] will put considerable strain on deployed WiMAX networks, a study of the proposed system for HD over WiMAX is reserved for future work. Short sequences such as *Paris* and *Football* were also selected for comparison with the work of others. These sequences are standard reference sequences chosen by the codec designers for their typicality and as a test of coding performance. Future work should also consider longer video streams or even carousels formed by a set of reference sequences to investigate further the effect of WiMAX network factors.

As previously mentioned, it is common for mobile devices to avoid the need for the more complex processing involved in bipredictive B-frames by using an IPPPP... Group of Pictures (GOP) structure. This arrangement also avoids sudden increases in latency [38] when periodic I-frames are employed. The frame rate was 30 Hz. It was necessary to protect against spatiotemporal error propagation in the event of inter-coded P-picture slices being lost. To ensure higher quality video, 2% intracoded MBs (randomly placed) were included in each frame (apart from the first I-picture) to act as anchor points in the event of slice loss. The JM 14.2 version of the H.264/AVC codec software was utilized to assess the objective video quality (Peak Signal-to-Noise Ratio (PSNR)) after packet loss, relative to the input YUV raw video. (YUV is not an acronym but the name of a color space that takes human perception of color into account.) In general, the configuration of the JM software is by a parameter file that acts as input to the decoder. Thus, this is how the percentage of randomly inserted intracoded MBs is specified.

Lost partition-C slice packets were compensated for by error concealment using the motion vectors in partition-A at the decoder to identify candidate replacement MBs in the last previously correctly received frame. Intra error concealment was also employed, as described below. In general, in the H.264/AVC codec standard, error concealment is a nonnormative feature, that is, a feature which is not needed for compliance with the standard. Nevertheless, in [39] a number of nonnormative error concealment algorithms for H.264/AVC were recommended, as, though error concealment is outside the scope of the standard, it is nevertheless needed. An attempt is made to conceal any lost slices. Error concealment within a lost slice is on an MB basis. Previously concealed MBs can be used to conceal missing MBs. Concealment proceeds from the edges of a lost slice inwards. For intracoded concealment of a missing MB, spatially adjacent pixels to a missing MB, if available, are interpolated to form the pixels of a missing MB. For inter-coded MBs within a lost slice, if very little motion has occurred, replacement by the matching MB in the previous frame occurs (known as error concealment by previous frame replacement). Otherwise, it is recommended [39] to

use one of the motion vectors of the surrounding MBs to identify a replacement MB. An algorithm to choose that motion vector is detailed in [39]. In the case of an MB split into subblocks, an average of the MVs of the subblocks within the MB is taken to form a candidate MV. The H.264/AVC algorithms will work even if only one correctly received slice is available within a frame.

A detailed guide to reconstruction of data-partitioned video by an H.264/AVC decoder is given in [40]. If partition-B is lost, missing MBs can be concealed by employing motion vectors from partition-A, and intra error concealment is optionally employed. In this sense, optional has a similar meaning to nonnormative, and in fact in the JM implementation used herein, intra error concealment is included. The procedure for lost partition-C packets has already been described. If both partition-B and partition-C go missing, then they are replaced by the MBs pointed to by the motion vectors in partition-A. If partition-A is lost, it is recommended to use the motion vectors of adjacent MB rows, that is, MBs from adjacent slices if these are available. Other ways of partitioning H.264/AVC coding data were also considered at the time of standardization such as splitting low and high transform coefficients normally present in partition-C and placing them in partition-A [41] or duplicating slice header and MB type information present in partition-A and placing it in partition-B [42]. The former [41] was recommended in certain circumstances when zigzag scanning of the transform coefficients is replaced by double scanning but, in terms of standardization, this recommendation appears to introduce "needless design variation" [43]. The latter [42] may introduce extra overhead [43] as the default case. Hence, [42] was also excluded from the standard.

It should also be remarked that others have conducted performance tests on using data partitioning. In [37] it was observed that it is possible to drop partitions B and C, while at the same time decreasing the quantization parameter (QP) to increase the video quality for an equivalent file size to retain the two partitions with a higher QP (lower video quality). However, this strategy was reported to only be worth trying for bipredictive B-frames, which in this current paper were not used. In [16], partition-A was repeated twice at low packet loss rates (3%), and three times at higher error rates (5, 10 and 20%) with competitive results compared to other forms of error resilience. In [44], in an approach that bears some resemblance to earlier work in [7], UEP was applied in an overlapping or sliding window fashion. In one experiment, each of the three partition types was aggregated from the frames in a GOP to form three segments. In another experiment, partition-As were combined with partition-Bs and accumulated as one segment, while the other segment was formed by aggregating all the partition-C NALUs within a GOP. Data from the anchor frame within the GOP was also included in the higher priority segment. The authors of [44] concluded that placing each data-partition type in its own segment was preferable to single-layer coding. It was also preferable to combine partition-A and partition-B, in terms of controlling the desired video data rate and erasure protection level.

Notice also that the JM implementation of random intracoded MBs does not duplicate placements of such MBs in previous frames, which was a defect identified in [45] of previous implementations of this form of intra placement. In fact, when all MB positions have been occupied over a sequence of frames, the random placement pattern is then repeated so that all MBs are refreshed in each cycle of the placement pattern. Therefore, the JM scheme, as it is a cyclic replacement one, can be compared to the use of a cyclic line of intracoded MBs. At best at the end of each cycle, all data is refreshed, and in that respect the use of randomly placed intracoded MBs acts just like the insertion of a periodic I-frame; that is, it provides a point of random access. However, the cyclic line procedure in CIF resolution frames refreshes at a quicker rate than the 2% of random intracoded used herein, as a horizontal line is equivalent to 5.5% of the MBs. For data-partitioned video this will lead to an increase in the size of partition-B and an increased bit-rate as a result. On the other hand, quality will on average be increased, not just due to the extra intracoded MBs but due to the fact that CIP will not restrict coding gain to such an extent. The latter gain arises as there are always adjacent intracoded MBs in a cyclic intra-refresh line. Therefore, future work can investigate the trade-offs between the different ways of inserting intracoded MBs.

4.3. *Rateless Decoder Modelling.* We used the following statistical model [46] to model the performance of the rateless decoder:

$$P_f(m, k) = 1 \quad \text{if } m < k$$

$$= 0.85 \times 0.567^{m-k} \quad \text{if } m \geq k, \quad (5)$$

where $P_f(m, k)$ is the decode failure probability of the code with k source symbols if m symbols have been successfully received (and $1 - P_f$ is naturally the success probability). Notice that the authors of [46] comment and show that for $k > 200$ the model of (5) almost perfectly models the performance of the code. This implies that if blocks are used approximately, 200 blocks should be received before reasonable behavior takes place. This observation also motivated the choice of bytes within a packet as the symbols, to reduce latencies. Upon receipt of the correctly received data, decoding of the information symbols is attempted, which will fail with a probability given by (5) for $k > 200$.

5. Evaluation

Tests evaluated various metrics, especially video quality for EEP and UEP alternatives. As mentioned in Section 3, in the UEP alternative partitions-A and -B form one segment with rateless coding applied, while partition-C was unprotected. The size of per-packet redundant data [5] was adaptively found from

$$R = \frac{L}{1 - BL} - L, \quad (6)$$

where L is the payload length and BL is the instantaneous probability of byte loss (a byte within a packet is the rateless

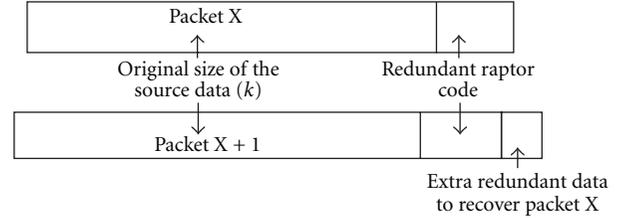


FIGURE 1: Division of payload data in a packet between source data, original redundant data, and extra redundant data for a previous erroneous packet.

code symbol). Up to 5% zero-mean Gaussian noise was additively included to distort the channel estimate in order to account for estimation inaccuracy. The rateless code belief propagation decoding algorithm has a small probability of failure and in which case extra redundant data were sent in the next packet. Only one retransmission over the WiMAX link is allowed to avoid increasing latency. However, as a retransmission request can be sent in the return TDD subframe, the additional delay is restricted to one WiMAX frame transmission time, that is, a minimum of 5 ms. Thus, if it turns out that the packet cannot be reconstructed, despite the provision of redundant data, extra redundant data are added to the next packet. In Figure 1, packet X is corrupted to such an extent that it cannot be reconstructed. Therefore, in packet X + 1 some extra redundant data is included up to the level that its failure is no longer certain. It is implied from (5) that if less than k symbols (bytes) in the payload are successfully received, then further $k - m + e$ extra redundant bytes can be sent to reduce the risk of failure. In the evaluation tests, e was set to four, resulting in a risk of failure of 8.7% (from (5)) in reconstructing the original packet if the extra redundant data successfully arrives. This reduced risk arises because of the exponential decay of the risk that is evident from (5) and which gives rise to Raptor code's low error probability floor [47].

To see the effect of channel conditions, the Gilbert-Elliott parameters were varied to produce a poor Channel 1 and a somewhat better Channel 2. The settings were CH1 = (PGG = 0.95, PBB = 0.96, PB = 0.02, PB = 0.165) and CH2 = (PGG = 0.97, PBB = 0.94, PB = 0.01, PB = 0.05). Similarly, the CBR data rate was tested both at 500 kbps and 1 Mbps for the two video clips of Section 4, *Football* and *Paris*. To ensure independence between partitions B and C, CIP was turned on, and 2% intra-refresh MBs were randomly added to the P-picture slices (refer to Section 4). Though a visual representation might pick out more clearly some results, for reasons of compactness and because some data representations are not helped by using charts, the presentation in this paper is through a set of tables.

5.1. *Results.* Tables 2 and 3 show EEP and UEP protection modes, respectively. No outright packet loss occurs in these and subsequent tables, except due to internal packet corruption, when attempts at packet repair have failed. Though the percentage of corrupted packets is high under

TABLE 2: Metrics for channel 1 with EEP.

With EEP	Football	Football	Paris	Paris
	2% IR	2% IR	2% IR	2% IR
	CIP	CIP	CIP	CIP
	500 kbps	1 Mbps	500 kbps	1 Mbps
	IPPP...	IPPP...	IPPP...	IPPP...
Dropped packets (%)	0	0	0	0
Packet end-to-end mean delay (s)	0.0068	0.0084	0.0068	0.0087
Mean PSNR (dB)	33.54	39.00	35.88	40.58
Corrupted packets (%)	24.61	30.64	21.77	30.55
Corrupted packet mean delay (s)	0.0170	0.0183	0.0166	0.0171

IR: intrarefresh, CIP: constrained intrarefresh.

TABLE 3: Metrics for channel 1 with UEP.

With UEP	Football	Football	Paris	Paris
	2% IR	2% IR	2% IR	2% IR
	CIP	CIP	CIP	CIP
	500 kbps	1 Mbps	500 kbps	1 Mbps
	IPPP...	IPPP...	IPPP...	IPPP...
Dropped packets (%)	11.02	10.38	12.77	15.11
Packet end-to-end mean delay (s)	0.0068	0.0083	0.0066	0.0080
Mean PSNR (dB)	30.56	30.5	28.3	28.02
Corrupted packets (%)	13.58	20.25	9.00	15.44
Corrupted packet mean delay (s)	0.0164	0.0183	0.0161	0.0170

IR: intrarefresh, CIP: constrained intrarefresh.

EEP, because extra redundant data for all partitions can be requested, it was possible to reconstruct all packets after one retransmission. However, under UEP, reconstruction of the longer partition-C packets was no longer possible, leading to an increase in the percentage of dropped packets to over 10% and a decrease in the percentage of corrupted packets, that is, packets that could be repaired. The main impact in terms of objective video quality (PSNR) is a drop in quality when UEP is employed.

Clearly, Table 3 shows the maximum drop in quality due to UEP, as it would also be possible to protect partition-C with a reduced percentage of rateless redundant data (rather than the zero percentage used). In contrast, gains from UEP are twofold. Firstly, because the percentage of corrupted packets is significantly reduced, the overall delay arising from the need to resend redundant data is reduced though mean corrupted packet delay is greater at 1 Mbps, as packets are longer. Secondly, under UEP there is an increase in the overall video bitrate arising from the reduction in rateless code overhead.

The mean per-frame overhead is given in Tables 4 and 5 for the *Football* and *Paris* sequences, respectively. The overhead from using UEP, in that respect, is about half of that of EEP. However, the maximum overhead for EEP at 500 kbps

TABLE 4: Mean per frame overhead in bytes from rateless coding for the *Football* sequence.

	Football 2% IR CIP	Football 2% IR CIP
	500 kbps IPPP...	1 Mbps IPPP...
EEP/CH1	41	84
UEP/CH1	28	51
EEP/CH2	10	20
UEP/CH2	7	12

IR: intrarefresh, CIP: constrained intraprediction.

TABLE 5: Mean per frame overhead in bytes from rateless coding for the *Paris* sequence.

	Paris 2% IR CIP	Paris 2% IR CIP
	500 kbps IPPP...	1 Mbps IPPP...
EEP/CH1	42	82
UEP/CH1	19	25
EEP/CH2	10	19
UEP/CH2	5	6

IR: intrarefresh, CIP: constrained intraprediction.

(42 B at 30 Hz) is a rate of $42 \times 8 \times 30 = 10$ kbps or 2% of the CBR rate. For EEP at 1 Mbps the maximum overhead is $84 \times 8 \times 30 = 20$ kbps or again 2% of the CBR rate. Therefore, the relative bitrate saving from using UEP rather than EEP is about 1% of the overall bitrate, which is obviously a small percentage. For this small gain in bitrate the drop in video quality is severe.

To investigate wireless channel dependency, results were taken for the channel 2 characteristics given in the introduction to this section. From Table 6, under EEP the performance metrics essentially remain the same as for channel 1, except for a reduction in the number of corrupted packets arising from the improved channel conditions. This will cause overall delay to be reduced but, as no packets are lost outright, there is no loss in video quality. When UEP is employed in Table 7, there is also a reduction in the percentage of dropped packets, in most cases to below 10%. This has the effect of improving the objective video quality by several dB but the quality is still well below the level of the EEP streams.

These results imply that in both types of channel conditions tested there is a significant negative impact on video quality from reducing protection of partition-C. As previously mentioned, motion-copy (MC) error concealment [39] is employed at the decoder to compensate for loss of partition-C. However, the gains from using MC error concealment to compensate for the loss of partition-C are not strongly apparent in the results. That observation can be applied to both the types of video content tested. This does not mean that there is no gain from data partitioning, as it has been long known that MC error concealment can significantly improve video quality in all but highly active video sequences. For example, in [21] there was a 5 dB improvement in quality from applying error concealment to MPEG-4 Part 2 data partitioning. In [48],

TABLE 6: Metrics for channel 2 with EEP.

With EEP	Football	Football	Paris	Paris
	2% IR	2% IR	2% IR	2% IR
	CIP	CIP	CIP	CIP
	500 kbps	1 Mbps	500 kbps	1 Mbps
	IPPP...	IPPP...	IPPP...	IPPP...
Dropped Packets (%)	0	0	0	0
Packet end-to-end mean delay (s)	0.0067	0.0084	0.0068	0.0082
Mean PSNR (dB)	33.54	39.00	35.88	40.58
Corrupted packets (%)	11.41	20.51	12.33	18.00
Corrupted packet mean delay (s)	0.0172	0.0180	0.0163	0.0169

IR: intrarefresh, CIP: constrained intraprediction.

TABLE 7: Metrics for channel 2 with UEP.

With UEP	Football	Football	Paris	Paris
	2% IR	2% IR	2% IR	2% IR
	CIP	CIP	CIP	CIP
	500 kbps	1 Mbps	500 kbps	1 Mbps
	IPPP...	IPPP...	IPPP...	IPPP...
Dropped packets (%)	3.71	6.79	7.55	11.11
Packet end-to-end delay (s)	0.0067	0.0081	0.0065	0.0079
Mean PSNR (dB)	32.22	34.9	30.76	30.16
Corrupted packets (%)	7.69	13.71	4.77	6.88
Corrupted packet mean delay (s)	0.0159	0.0179	0.0156	0.0164

IR: intrarefresh, CIP: constrained intraprediction.

the gain after whole frame loss from refining motion-copy (RMC) error concealment (through recursive estimation of motion vectors over multiple frames) was found to improve over previous frame replacement (PFR) and MC error concealment.

In [49], for a 5% packet loss rate, MC of the motion vectors of the last reference frame improved upon PFR by at least 2 dB in PSNR, and a further 2 dB at least if RMC was used. Conversely, the availability of the correct motion vectors from protected partition-A (rather than estimated ones) will significantly benefit video quality. It should also be added that, for broadcast quality video, the smaller partition-A packet sizes [6] are an additional form of protection relative to larger partition-C packets, even when EEP is applied.

6. Conclusion

As user expectations of mobile video streaming increase, video quality becomes an important determinant of the take-up of a service. In this paper, it was shown that equal error protection can result in several dBs gain in video quality over unequal error protection of data-partitioned video. The overhead from using EEP rather than UEP was about

1% of the overall constant bit rate. Consequently, as data-partitioning already brings advantages in terms of smaller packet sizes for more important data and the ability to compensate if texture data is lost, equal error protection is preferable, except when there is a severe shortage of available bandwidth. As the recent trend is towards much greater bandwidth capacity for mobile systems, then the bitrate savings from UEP may no longer be worth pursuing.

There are a number of avenues for future research. Section 4.2 mentioned the need for testing the scheme with the emerging HD and 3D (stereoscopic) resolutions that will eventually migrate to broadband wireless streaming. That section also mentioned alternative ways of partitioning H.264/AVC coding data, which was not standardized but nevertheless was worth considering. It is also possible to propose still other ways of subpartitioning partitions B and C, which have been investigated by some of the authors. The merits of these schemes and different forms of packetization are worthy of investigation. In Section 4.2 also, it was mentioned that there are many alternatives for insertion of intracoded MBs, which will each have their effect on the resulting video quality. This paper has considered CBR video but Variable Bit Rate (VBR) video is often preferred by researchers, because, despite time-varying data rates and unpredictable storage requirements, it results in an even quality. By virtue of open-loop coding, it also results in a simpler codec. In particular, CBR is unsuitable for HD video as quality variations are more apparent. This suggests that at a cost in delay the impact of the protection scheme for smoothed HD video streaming should be investigated. For VBR video streams, varying the QP will impact on the distribution of coding data between partitions. Therefore, the impact of QP dependency on the robustness of the scheme can also be investigated.

References

- [1] D. Raychaudhuri and N. B. Mandayam, "Frontiers of wireless and mobile communications," *IEEE Proceedings*, vol. 100, no. 4, pp. 824–840, 2012.
- [2] J. G. Andrews, A. Ghosh, and R. Muhammed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, Prentice Hall, Upper Saddle River, NJ, USA, 2007.
- [3] T. Stockhammer, P. Fröjdth, I. Sodagar, and S. Rhyu, "Information technology—MPEG systems technologies—Part 6: Dynamic adaptive streaming over HTTP (DASH)," ISO/IEC MPEG Draft International Standard, 2011.
- [4] H. Kalva, V. Adzic, and B. Furht, "Comparing MPEG AVC and SVC for adaptive HTTP streaming," in *Proceedings of the IEEE International Conference on Consumer Electronics*, pp. 160–161, 2012.
- [5] 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)*, p. 6, June 2010.
- [6] L. Al-Jobouri, M. Fleury, and M. Ghanbari, "Error-resilient IPTV for an IEEE 802.16e channel," *Wireless Engineering and Technology*, vol. 2, no. 2, pp. 70–79, 2011.
- [7] 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 IEEE International Conference on Image Processing (ICIP '09)*, pp. 909–912, November 2009.
- [8] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [9] T. Stockhammer and M. Bystrom, “H.264/AVC data partitioning for mobile video communication,” in *Proceedings of the International Conference on Image Processing (ICIP '04)*, pp. 545–548, October 2004.
- [10] S. Mys, P. Lambert, and W. De Neve, “SNR scalability in H.264/AVC using data partitioning,” in *Proceedings of the Pacific Rim Conference on Multimedia*, pp. 329–338, 2006.
- [11] O. I. Hillestad, A. Perkis, V. Genc, S. Murphy, and J. Murphy, “Adaptive H.264/MPEG-4 SVC video over IEEE 802.16 broadband wireless networks,” in *Proceedings of the 16th International Packet Video Workshop*, pp. 26–35, November 2007.
- [12] 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.
- [13] H. H. Juan, H. C. Huang, C. Huang, and T. Chiang, “Cross-layer mobile WiMAX MAC designs for the H.264/AVC scalable video coding,” *Wireless Networks*, vol. 16, no. 1, pp. 113–123, 2008.
- [14] J. Casasepere, P. Sanchez, T. Villameriel, and J. Del Ser, “Performance evaluation of H.264/MPEG-4 scalable video coding over IEEE 802.16e networks,” in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '09)*, pp. 1–6, May 2009.
- [15] T. Stockhammer, “Is fine granular scalable video coding beneficial for wireless video applications?” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 193–196, 2003.
- [16] S. Wenger, “H.264/AVC over IP,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [17] M. M. Hannuksela, Y. K. Wang, and M. Gabbouj, “Isolated regions in video coding,” *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 259–267, 2004.
- [18] C. Bormann, C. Burmeister, M. Degermark et al., “RObust Header Compression (ROHC): framework and four profiles: RTP, UDP, ESP, and uncompressed,” IETF RFC 3095, 2001.
- [19] Y. Dhondt, S. Mys, K. Vermeirsch, and R. Van de Walle, “Constrained inter prediction: removing dependencies between different data partitions,” in *Advanced Concepts for Intelligent Visual Systems*, pp. 720–731, 2007.
- [20] C. M. T. Calafate and M. P. Malumbres, “Evaluation of the H.264 codec,” Tech. Rep. DISCA-UPV-2003, Universidad Politecnica Valencia, 2003.
- [21] W. Rabiner, M. Budagavi, and R. Talluri, “Proposed extensions to DMIF for supporting unequal error protection of MPEG-4 video over H.324 mobile networks,” in *ISO/IEC JTC 1/SC 29/WG 11, Doc. M4135, MPEG Atlantic City Meeting*, October 1998.
- [22] S. T. Worrall, S. N. Fabri, A. H. Sadka, and A. M. Kondoz, “Prioritisation of data partitioned MPEG-4 video over mobile networks,” *European Transactions on Telecommunications*, vol. 12, no. 3, pp. 169–174, 2001.
- [23] B. Barmada, M. M. Ghandi, E. V. Jones, and M. Ghanbari, “Prioritized transmission of data partitioned H.264 video with hierarchical QAM,” *IEEE Signal Processing Letters*, vol. 12, no. 8, pp. 577–580, 2005.
- [24] G. H. Yang, D. Shen, and V. O. K. Li, “UEP for video transmission in space-time coded OFDM systems,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, pp. 1200–1210, March 2004.
- [25] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, “Priority encoding transmission,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1737–1744, 1996.
- [26] A. E. Mohr, E. A. Riskin, and R. E. Ladner, “Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 819–828, 2000.
- [27] M. Luby, T. Stockhammer, and M. Watson, “IPTV systems, standards and architectures: part II—application layer FEC In IPTV services,” *IEEE Communications Magazine*, vol. 46, no. 5, pp. 94–101, 2008.
- [28] ITU-T Rec. Y.1541, “Internet Protocol Aspects—Quality of Service and Network Performance: Network Performance Objectives for IP-Based Services,” 2002.
- [29] ETSI TS 102 034 v1.3.1, “Transport of MPEG 2 Transport Stream (TS) Based DVB Services over IP Based Networks,” DVB Blue Book A086rev5, October 2007.
- [30] 3GPP TS26.346, “Multimedia Broadcast/Multicast Service (MBMS): Protocols and Codecs,” December 2005.
- [31] M. Luby, “LT codes,” in *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, November 2002.
- [32] F. C. D. Tsai, J. Chen, C.-W. Chang, W.-J. Lien, C.-H. Hung, and J.-H. Sum, “The design and implementation of WiMAX module for ns-2 Simulator,” in *Proceedings of the Workshop on ns2: The IP Network Simulator (WNS2 '06)*, article no. 5, 2006.
- [33] IEEE, “802.16e-2005, IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems,” 2005.
- [34] C. Jiao, L. Schwiebert, and B. Xu, “On modeling the packet error statistics in bursty channels,” in *Proceedings of the IEEE Conference on Local Computer Networks*, pp. 534–541, 2002.
- [35] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, “Analysis of packet loss for compressed video: effect of burst losses and correlation between error frames,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 861–874, 2008.
- [36] D. Marpe, T. Wiegand, and S. Gordon, “H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 593–596, September 2005.
- [37] H. Bing, *3D and HD Broadband Video Streaming*, Artech House, Boston, Mass, USA, 2010.
- [38] R. M. Schreier and A. Rothermel, “Motion adaptive intra refresh for low-delay video coding,” in *Proceedings of the International Conference on Consumer Electronics (ICCE '06)*, pp. 453–454, January 2006.
- [39] V. Varsa, M. Hannuksela, and Y. K. Wang, “Non-normative error concealment algorithms,” in *Proceedings of the 14th Meeting of ITU-T Video Coding Experts Group*, 2001, doc.VCEG-N62.
- [40] S. Wenger and T. Stockhammer, “H.26L over IP and H.324 framework,” in *Proceedings of the 14th Meeting of ITU-T Video Coding Experts Group*, Santa Barbara, Calif, USA, 2001, doc. VCEG-N52.
- [41] J. C. Ye and Y. Chen, “Flexible data partitioning mode for streaming video,” in *Proceedings of the 4th Meeting of Joint Video Team (JVT '02)*, July 2002, doc. JVT-D136.

- [42] T. Stockhammer, "Independent data partitions A and B," in *Proceedings of the 3rd Meeting of Joint Video Team (JVT '02)*, May 2002, doc. VT-C132.
- [43] G. Sullivan, "Seven steps toward a more robust codec design," in *Proceedings of the 3rd Meeting of Joint Video Team (JVT '02)*, May 2002, doc. JVT-C117.
- [44] S. Nazir, D. Vukobratovic, and V. Stankovic, "Expanding window random linear codes for data partitioned H.264 video transmission over DVB-H network," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 2205–2208, 2011.
- [45] G. Côté and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," *Signal Processing: Image Communication*, vol. 15, no. 1, pp. 25–34, 1999.
- [46] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 235–246, 2007.
- [47] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," in *Proceedings of the IEEE International Symposium on Information Theory*, p. 37, July 2004.
- [48] J. T. Chien, G. L. Li, and M. J. Chen, "Effective error concealment algorithm of whole frame loss for H.264 video coding standard by recursive motion vector refinement," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1689–1695, 2010.
- [49] S. K. Bandyopadhyay, Z. Wu, P. Pandit, and J. M. Boyce, "An error concealment scheme for entire frame losses for H.264/AVC," in *Proceedings of the IEEE Sarnoff Symposium*, pp. 1–4, March 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

