Effective Video Transport over WiMAX with Data Partitioning and Rateless Coding

Laith Al-Jobouri, Martin Fleury, Salah S. Al-Majeed, and Mohammed Ghanbari
School of Comput. Sci. and Electron. Eng.
University of Essex
Colchester, United Kingdom
{lamoha, fleum, ghan}@essex.ac.uk

***Abstract*—Video streaming is anticipated to be a key application of broadband wireless access networks such as WiMAX. This paper proposes a combination of data-partitioning of compressed video information and rateless channel coding to ensure effective video transport. A counter-intuitive result is that comparatively improved objective video quality occurs even though privileged application-layer forward error correction is *not* given to high priority data. Instead a flat channel coding is used across the data partitions. The scheme results in a lower number of dropped packets at the transmitter buffer and/or a reduced number of packets corrupted by channel noise compared to simple slicing or no frame slicing at all. Larger-sized IEEE 802.16 (WiMAX) Time Division Duplex frames are found to reduce the number of packets dropped through traffic congestion.**

Keywords-broadband wireless; rateless channel coding; error resilience; source coding; video streaming, WiMAX

I. INTRODUCTION

IEEE 802.16e (mobile WiMAX) [1] provides broadband wireless access independent of a pre-existing cellular system, is not dependent on hardware authentication, can deliver data in a cost-effective way at 3-4 times the rate of 3G cellular systems, and is currently deployed, rather than in development. Its main technological weakness may be that it uses Orthogonal Frequency Division Multiple Access (OFDMA) for both the uplink and downlink transmission, rather than OFDMA for the downlink and Single Carrier-Frequency Division Multiple Access (SC-FDMA), which confers power saving advantages on Long Term Evolution (LTE) [2]. WiMAX is suited to provide dedicated multimedia services, with existing services in Brazil [3] and Korea [4] (as WiBro is now harmonized with WiMAX).

In this paper, we develop an effective video streaming system for WiMAX that provides error resilience through data partitioning [5] and which works in this paper *without* the need to apply privileged protection to the high-priority partitions. This is achieved in the H.264/AVC (Advanced Video Coding) codec [6] by setting the quantization parameter (QP) in such a way that lower-priority texture data that can be replaced more easily at the decoder through error concealment occupies a larger part of a frame's data. Consequently, when packetized in a WiMAX MAC Service Data Unit (MSDU) within a MAC Protocol Data Unit (MPDU) [7] it is more likely to suffer error than MSDUs bearing data from other partitions.

However, it is still necessary to protect the whole of the bitstream (without privileging the A- and B-partitions) against the risk of packet loss. Application-layer rateless channel coding [8] was selected for its flexibility and its linear computational complexity at both the decoder and the encoder. To avoid long latencies, which would occur if packet-level forward error correction (FEC) were to be applied, redundant data was added to packets themselves, treating the bytes within each packet as the data symbols. Again to reduce latency, a single Automatic Repeat request (ARQ)s was made if the available redundant data were insufficient to reconstruct a corrupted packet.

Configuration of the WiMAX Time Division Duplex (TDD) frame size is also important as this size governs the service time given to each service queue (assuming video is allocated to the real-time Polling Service (rtPS) class of service). In [9] for real-time streaming over WiMAX, the WiMAX MPDU size was varied depending on the channel state estimation. At the same time FEC is applied but again the FEC overhead is dynamically varied. However, this paper does not consider source coding aspects of streaming. On the other hand, research in [10] does use the Scalable Video Coding extension of the H.264 codec but takes little account of channel conditions, which pose a high risk to base layer packets, as these can lead to the discard of dependent packets in other layers.

The following Section now establishes the context of the scheme before the paper goes on to develop the simulation model and evaluate the proposal in Section III. Section IV concludes the paper with a discussion of the implications and further research.

II. BACKGROUND

A. Data partitioning

The H.264/AVC codec conceptually separates the Video Coding Layer (VCL) from the Network Abstraction Layer (NAL). The VCL specifies the core compression features, while the NAL supports delivery over various types of network. This network-friendliness feature of the standard

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

facilitates easier packetization and improved video delivery. In addition, to adapt H.264 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, allows communication over a variety of different channels.

The Network Abstraction Layer (NAL) facilitates the delivery of the H.264 VCL data to the underlying transport layers such as RTP/IP, H.32X and MPEG-2 systems. 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 extensions and the types 24 to 31 are unspecified.

In the H.264 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 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 Data Partitioning (DP) mode is active, as now explained.

In type 1 and type 5 NALs, MB addresses, motion vectors and the transform coefficients of the blocks, are packed into the packet, in the order as 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 motion vectors are much more important than the (integer) 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 motion vectors (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, this is known as data partitioning (DP).

In H.264, 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. 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, motion vectors and essential headers. If any MBs in these pictures are intra-coded, 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.

Fig. 1 is a comparison between the relative sizes of the partitions according to QP for two diverse reference video clips. In H.264/AVC the QP range is from 0 to 51, with a low QP representing high quality. The Paris sequence is a studio scene with two head and shoulders images of presenters. The background is of high spatial complexity. In contrast, 'Stefan' is a tennis-playing sequence representing rapid motion, and thus high temporal complexity. Both clips were encoded at Common Intermediate Format (CIF) (352×288 pixel/frame), with a Group of Picture (GOP) structure of IPPP….. at 30 Hz. Experiments not shown indicate that including B-Pictures, with a GOP structure of IPBP (sending order) … and intra-refresh rate of 15, did not noticeably disturb this pattern.

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

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

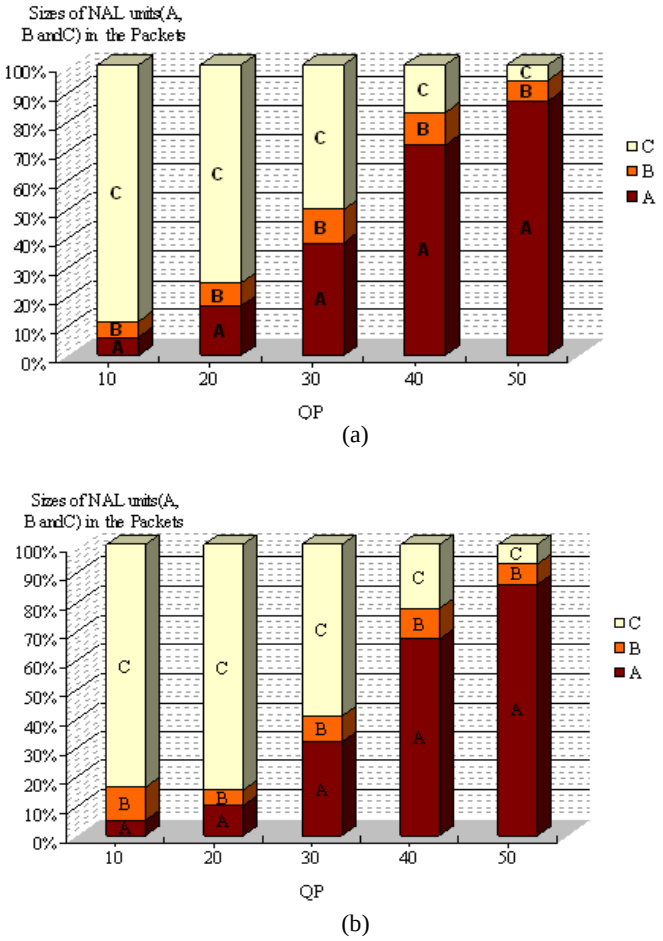

Figure 1. Relative sizes of data partitions according to quantization parameter (QP) for (a) Paris, (b) Stefan video sequence.

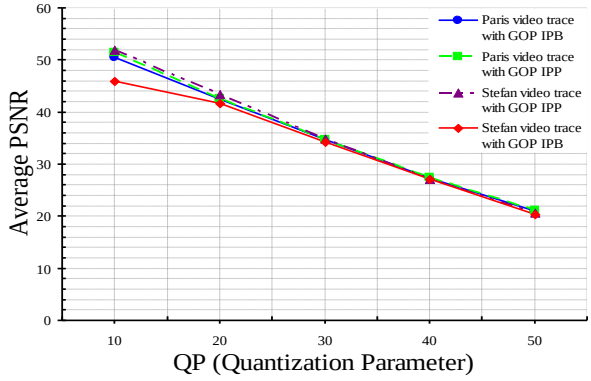

Figure 2. Average PSNR with different QP for the Paris and Stefan sequences with different GOP structures.

Clearly the relatively small size of the A- and B-partitions is a potential advantage at high 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 afforded A partition NALs, then they become relatively vulnerable to packet loss by virtue of their relatively increased length. From Fig. 2, setting the QP value at the high end of the range results in unacceptable video quality even for mobile applications (PSNR below 25 dB). However, video quality at the high-end is above 40 dB, which for many purposes at CIF resolution is unnecessary.

B. Rateless coding

Rateless coding is ideally suited [11] 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) Reed-Solomon (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, for RS coding 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 [11] 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 [12] 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 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 [13]) 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 [13] is a decoding complexity that is linear in k. A systematic Raptor code is arrived at [13] by first applying the inverse of the inner code to the first k symbols before the outer pre-coding step.

In order to model Raptor coding, we employed the following statistical model [14]:

$$P_f(m,k) = 1 \quad \text{if} \quad m < k,$$
$$= 0.85 \times 0.567^{m-k} \text{ if } m \geq k \qquad (1)$$

where $P_f(m,k)$ is the failure probability of the code with k source symbols if m symbols have been received. The authors of [14] remark and show that for $k > 200$ the model almost perfectly models the performance of the code. When $m = k$ with a failure rate of as much as 85%, but this failure rate reduces exponentially as the difference between m and k grows. In the experiments reported in this paper, the percentage redundancy for the Raptor code was set to 10% similarly to the usage in [15] for video streaming. The symbol size was set to bytes within a packet. Clearly, if instead 200 packets are accumulated before the rateless decoder can be applied (or at least equation (1) is relevant) there is a penalty in start-up delay for the video streaming and a cost in providing sufficient buffering.

A corrupt packet can be detected by the Cyclic Redundancy Check (CRC) that is an optional part of the MPDU (WiMAX packet), refer to Fig. 3. Though this CRC also applies to the 4-byte MAC header, it does indicate the likelihood that a packet's payload is corrupt. Then, through measurement of channel conditions, an estimate of the number of symbols successfully received is made, giving a value *m'*. This implies from (1) that if less than k symbols (bytes) in the payload are successfully received then *k-m'+1* redundant bytes can be sent to reduce the risk of failure to below 50%. Clearly, it is possible to tune the failure risk by simply including more bytes. However, in this paper we confine repeat transmissions of redundant bytes to a minimal amount. To reduce latency, the number of retransmissions, after an ARQ over the uplink, is limited to one. In Fig. 4, 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.

WiMAX already specifies [16] that a SS should provide channel measurements that can form a basis for channel quality estimates. These are either Received Signal Strength Indicators or may be Carrier-to-Noise-and Interference Ratio measurements made over modulated carrier preambles.

III. SIMULATION MODEL

A. *Channel model*

To establish the behavior of rateless coding under WiMAX the well-known ns-2 simulator was augmented with a module [17] that has proved an effective way of modeling IEEE 802.16e's behavior.. We also introduced a two-state Gilbert-Elliott channel model [18] in the physical layer of the simulation to simulate the channel model for WiMAX. The PGG (probability of being in a good state) was set to 0.95, PBB (probability of being in a bad state) = 0.96, PG (probability of losses in a bad state) = 0.02 and PB (probability of losses in a bad state) = 0.165 for the Gilbert and Elliott parameters in Section IV. As an illustration of the effect, the PB was increased by 0.03 until 0.3. Four different 30 frame/s (Hz) video data-partitioning traces with Common Intermediate Format (CIF) spatial resolution of 352 ×288 pixel/frame were utilized. The packet loss percentage was calculated to find the relationship between the lost packets and PB. From Fig. 5, it is apparent that as much as the PB is increased the packet loss percentage is increased, as might be expected. However, because of the different coding complexities and types of complexity (spatial and temporal) the effect on video quality will be different.

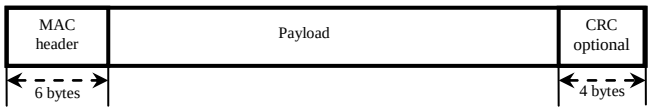

Figure. 3 General format of a MAC PDU.

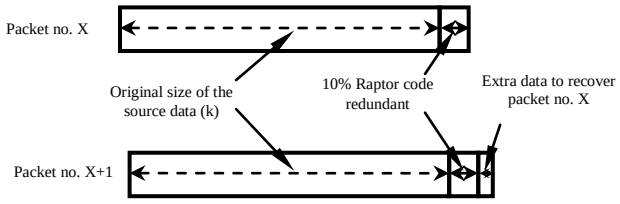

Figure 4. Division of payload data in a packet (MPDU) between source data, original redundant data and piggybacked data for a previous errored packet.

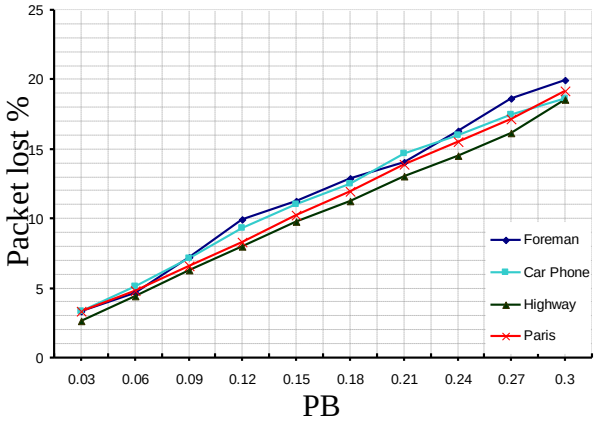

Figure 5. The relationship between Gilbert-Elliott bad probability (PB) and packet loss rate for different video sequences.

B. *WiMAX configuration*

The physical layer (PHY) settings selected for WiMAX simulation are given in Table II. The antenna is modeled for comparison purposes as a half-wavelength dipole. The Gilbert-Elliott 'bursty' channel model is further explained in Section 2.5. The TDD frame length was varied in experiments, because, as mentioned in Section 1, it has an important effect on the service rate at an SS.

Video was transmitted over the downlink with UDP transport. In order to introduce sources of traffic congestion, an always available FTP source was introduced with TCP transport to the SS. Likewise a CBR source with packet size of 1000 B and inter-packet gap of 0.03 s was downloaded to the SS. While the CBR and FTP occupy the non-rtPS queue, rather than the rtPS queue, they still contribute to packet drops in the rtPS queue for the video, if too many video packets occupy the 50 packet buffer, while the nrtPS queue is being serviced.

C. Video configuration

The Paris sequence, mentioned in Section II.A was employed for the WiMAX downlink tests. Paris with 1064 CIF frames was Variable Bit Rate (VBR) encoded at 30 Hz. As a GOP structure of IPPP.... was employed it is necessary to protect against error propagation in the event of intra-coded P-frame slices being lost. Gradual Decoder Refresh (GDR) from H.264/AVC inserts 25% inter-coded macroblocks (randomly placed) to act as anchor points in the event of slice loss. The advantgage of the GOP configuration is that it allows H.264/AVC's baseline profile to operate with reduced codec complexity due to the absence of Bi-predictive B-frames. At the decoder motion copy error concealment was set, allowing the motion vectors contained in A-partition packets to indicate suitable replacement macroblocks within the last correctly received slice. The JM 14.2 version of the codec software was employed with the Evalvid environment used to reconstruct sequences according to reported packet loss from the simulator.

TABLE II. IEEE 802.16 PARAMETER SETTINGS

Parameter	*Value*
PHY	OFDM
Frequency band	5 GHz
Bandwidth capacity	10 MHz
Duplexing mode	TDD
Frame length	8 to 20 ms
Max. packet length	1024 B
Raw data rate	10.67 Mbps
IFFT size	1024
Modulation	16-QAM 1/2
Guard band ratio	1/8
DL/UL ratio	3:1
Channel model	Gilbert-Elliott
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, QAM = Quadrature Amplitude Modulation, TDD = Time Division Duplex

IV. EVALUATION

Two types of erroneous packets were considered: 1) packet drops at the BS sender buffer and 2) corrupted packets that were received but affected by Gilbert-Elliott channel noise. The Raptor code equation (1) was then applied to decide if a packet could be recovered, given the number of bytes that were declared to be in error. Tests first checked whether partitioning a frame into three was responsible for the gain from data-partitioning or whether the gain actually arose from the prioritized segregation of the information when using DP. The comparison was made across QP and across WiMAX TDD frame sizes. Therefore, in the non-data-partitioning (non-DP) sliced tests, an encoded frame was geometrically divided into three equal-sized horizontal slices (simple slicing).

Comparing Tables III and IV, it is apparent that geometric slicing has an advantage in the number of dropped packets for high quality video (QP = 10). Otherwise, data-partitioning is prefereable, because the realively fewer A-partition packets corrupted allows better quality decoder reconstructions. The small fluctuations in the number of geometrically-sliced corrupted packets makes little difference to the objective video quality fro QPs higher than 10. One can conclude that more C-partition packets than A- and B- packets are corrupted, leading to the superior quality of the data-partioned solution, even though no special protection is given to A- and B-partition packets. Looking at the variation in WiMAX TDD frame size, it is in relation to a reduction in dropped packets at the BS buffer that using the larger frame size brings benefits. Though commercial settings are difficult to establish it could be that a low TDD frame size as small as 5 ms could be common.

A further comparison was made with the effect of not employing data-partitioning or slicing. In other words, the encoded video was configured with one slice per frame. If the single slice was larger than the maximum packet size of 1024 B then it is segmented at the network level. This also could occur with data-patitioning or slicing but is less likely. From Table V, it is apparent that effectively increasing the packet size leads at low QP to huge numbers of dropped packets. This is because a lower QP setting leads to larger slices emerging from the encoder. The effect of the high number of dropped packets is that the decoder was unable to decode the video when multiple successive packets were dropped. However, as the percentage of corrupted packets was also over 10% in many tests, the objective video quality was also lower than when data-partioning was employed. This was the case even though rateless channel coding was deployed.

Though other tests were performed to check the performance of data-partitioned packetization without Raptor code, the high loss rate of A-partition packets prevented the decoder reconstructing the video. The packet end-to-end delay was also measured. Considering the delay for the data-partitioning scheme in Table VI, this tends to reduce with lower packet size and higher QP. There is also a reduction with longer TDD frame size. The first effect is the result of on average longer transmission times, while the latter effect can be explained by reduced queueing times. For single slice per

TABLE III. LOST PACKETS AND VIDEO QUALITY WHEN USING DATA-PARTITIONING WITH RAPTOR CODE.

	Dropped packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.049	0.014	0.001
15	0	0	0
20	0	0	0
25	0	0	0
	Corrupted packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.096	0.097	0.093
15	0.095	0.092	0.098
20	0.088	0.096	0.093
25	0.088	0.093	0.099
	PSNR (dB)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	26.48	34.43	42.26
15	46.95	46.95	46.95
20	42.98	42.98	42.98
25	39.30	39.30	39.30

TABLE IV. LOST PACKETS AND VIDEO QUALITY WHEN USING GEOMETRIC SLICING WITH RAPTOR CODE.

	Dropped packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0	0	0
15	0	0	0
20	0	0	0
25	0	0	0
	Corrupted packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.087	0.088	0.086
15	0.088	0.088	0.090
20	0.094	0.089	0.081
25	0.084	0.081	0.086
	PSNR (dB)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	51.19	51.19	51.19
15	42.24	42.24	42.24
20	42.53	42.53	42.53
25	39.02	39.02	39.02

TABLE V. LOST PACKETS AND VIDEO QUALITY WHEN USING SINGLE SLICES WITH RAPTOR CODE.

	Dropped packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.998	0.998	0.998
15	0.783	0.664	0.282
20	0.001	0	0
25	0	0	0
	Corrupted packets (%)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	-	-	-
15	0.112	0.103	0.104
20	0.111	0.100	0.080
25	0.103	0.103	0.097
	PSNR (dB)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	-	-	-
15	-	-	27.94
20	42.10	42.53	39.01
25	39.01	39.01	39.01

TABLE VI. MEAN PACKET END-TO-END DELAY FOR THE SCHEMES COMPARED.

	Data partitioning packet end-to-end delay (s)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.072	0.064	0.053
15	0.019	0.025	0.029
20	0.012	0.019	0.022
25	0.0092	0.016	0.020
	Simple slicing packet end-to-end delay (s)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	0.008	0.015	0.019
15	0.008	0.015	0.019
20	0.008	0.015	0.019
25	0.008	0.015	0.019
	Single slice frames packet end-to-end delay (s)		
QP	8 ms TDD	10 ms TDD	20 ms TDD
10	-	-	-
15	0.132	0.124	0.100
20	0.024	0.029	0.032
25	0.016	0.022	0.025

frame this pattern was repeated but the latency times were increased as a result of the larger packets. Simple slicing does not follow this pattern because there are no larger packets to weigh the delay averages upwards. However, it is unclear why there is little variation dependency on QP and hence slice size and why queuing appears to increase with larger TDD frame size, except that jitter within the video stream tends to be reduced. For the sliced schemes (data-partitioning and simple slicing, end-to-end delay is generally below 30 ms, except for the highest quality video transported with data-partitioning. Therefore, for these schemes jitter buffers at the SS can be small.

V. CONCLUSIONS

In the proposed scheme for WiMAX downlink video streaming, the compressed video bitstream is partitioned according to the information's importance to the decoding process. It was found that, provided rateless coding is used, data-partitioning gains over simple geometric slicing or no slicing within a frame. This is because more C-partition packets tend to be lost because of their larger size, while the preservation of A- and B-partition slices still allows reconstruction through motion error concealment at the decoder. Differential protection through a form of windowed rateless coding could further improve the quality but this would be at a cost in complexity. Further work will investigate optimal settings for the ratio of rateless coding data and additional data sent after an initial failure to reconstruct a packet. The adaptive rateless coding scheme can also be extended.

REFERENCES

[1] 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.

[2] K. Ekstrom et al., "Technical solutions for the 3G Long-Term Evolution," IEEE Commun. Mag., vol. 44, no. 3, 2, pp. 38-45, 2006.

[3] L. G. P. Meloni, "A new WiMAX profile for DTV return channel and wireless access," in Mobile WiMAX, Chen, K.-C. and de Marca, J. R. B. Eds., Wiley & Sons, Chichester, UK, 2008, pp. 291-392,

[4] H. Kim, J. Lee, and B. G. Lee, "WiBro – A 2.3 GHz MobileWiMAX: System design, network deployment, and services," in *Mobile WiMAX*, Chen, K.-C. and de Marca, J. R. B. (eds.), Wiley & Sons, Chichester, UK, 2008, pp. 257-290.

[5] S. Wenger, "H.264/AVC over IP," IEEE Trans. Circuits and Syst. for Video Technol., vol. 13, no. 7, pp. 645-655, 2003.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, July 2003.

[7] M. Fleury, R. Razavi, S. Saleh, L. Al-Jobouri, and M. Ghanbari, "Enabling WiMAX video steaming," in WiMAX, New Developments, In-Tech, Vukovar, Croatia, 2009, pp. 213-238.

[8] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video streaming over MBMS: A system design approach," J. of Multimedia, vol. 1, no. 5, pp. 25-35, 2006.

[9] M. Chatterjee, S. Shengupta, and S. Ganguly, "Feedback-based real-time streaming over WiMAX," IEEE Wireless Communicaations, vol. 14, no. 1, pp.64-71, 2007.

[10] H.-H. Juan, H.-C. Huang, C.-Y. Huang, and T. Chiang, "Scalable video streaming over mobile WiMAX," IEEE Int. Symp. Circ. Syst., pp. 3463-3466, 2007.

[11] D. J. C. MacKay, "Fountain codes," IEE Proceedings: Communications, vol. 152, no. 6, pp. 1062–1068, 2005.

[12] M. Luby, "LT codes," in Proc. of the 34rd Annual IEEE Symp. on Foundations of Computer Science, pp. 271–280, Vancouver, Canada, N ov. 2002.

[13] A. Shokorallahi, "Raptor codes," IEEE Trans. on Information Theory, vol. 52, no. 6, pp. 2551-2567, 2006.

[14] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," IEEE Trans. Broadcasting, vol. 53, no. 1, pp. 235-246, 2007.

[15] S. Ahmad_, R. Hamzaoui, M. Al-Akaidi, "Robust live unicast video streaming with rateless codes", *Int. PacketVideo Workshop*, Nov. 2007.

[16] L. Nuaymi, WiMAX: Technology for Broadband Wireless Access, J. Wiley & Sons Ltd, Chichester, UK, 2007.

[17] F. C. D. Tsai, et al., "The design and implementation of WiMAX module for ns-2 simulator," Workshop on ns*2*, article no. 5, 2006.

[18] G. Haßlinger and O. Hohlfeld, "The Gilbert-Elliott model for packet loss in real time services on the Internet," 14th GI/ITG Conf. on Measurement, Modelling, and Evaluation of Computer and Commun. Systs., pp. 269-283, 2008.