Adaptive Raptor Coded Video Streaming for a WiMAX Channel

Laith Al-Jobouri and Martin Fleury
School of CSEE
University of Essex
Colchester, U.K.
{lamoha, fleum}@essex.ac.uk

Abstract— **This paper proposes an adaptive channel coding scheme for data-partitioned video streaming over a mobile access network, namely IEEE 802.16e (WiMAX). Raptor coding, for reduced latency, is applied at the byte-level rather than the block level. Adaptation is achieved by a single retransmission, when necessary, of extra redundant data to reconstruct corrupted packets. Packet drops are responded to through the error resiliency inherent in partitioned source coding. The paper finds that operation of the proposed scheme depends on: choice of group-of-pictures structure; selection of an appropriate quantization parameter; and configuration of data partitioning to reduce inter-partition dependencies.**

Keywords— broadband wireless access; Raptor channel coding; video streaming

I. INTRODUCTION

Video streaming is anticipated to be a key application [1] of broadband wireless access networks such as IEEE 802.16e (mobile WiMAX) [2]. WiMAX is now being deployed in rural areas and in developing countries with a limited 3G infrastructure. However, 'bursty' errors can still disrupt a fragile compressed video bitstream. This is because of source coding data dependencies, which arise both from motion-compensated prediction and entropy coding. Consequently, those videos with high source-coding complexity are at risk. They have larger packet sizes and their pictures are difficult to reconstruct through error concealment, whenever prior or neighboring data are missing.

Data partitioning, which is a form of layered error resilience, can provide graceful degradation of video quality and as such [3] has found an application in mobile video streaming. In an H.264/AVC (Advanced Video Coding) codec, when data-partitioning is enabled, every slice is divided into three separate partitions: partition-A (has the most important data, motion vectors); -B (intra coefficients); and -C (inter coefficients, with the least important data). These data are packed into three types of Network Abstraction Layer units (NALU's), prior to encapsulation in RTP/IP/UDP packets. The importance of each NALU-bearing packet is identified in the NALU header.

A key issue for data-partitioned streaming is how to provide protection against a harsh channel environment. In this paper, a simplified approach is to protect all partitions equally but rely on the smaller sizes of partition-A and –B carrying packets. However, the quantization parameter (QP) should be chosen to ensure the smaller sizes. As a result of the smaller packet sizes, there is a lower risk of channel error and buffer overflow. When one or both of the higher priority partitions are received successfully then error concealment allows reconstruction of the video slice.

The paper also caters for packets that are corrupted but may still be repairable by Raptor channel coding [4]. Raptor coding has O(n) decoding computational complexity, which should be compared to the $O(n^3)$ complexity of some codes. A point to note is that such rateless codes are a probabilistic channel code, in the sense that reconstruction is not guaranteed. Prior use of application-layer Raptor coding has been in various wireless standards [5]. However, that usage has been through multicast [6] rather than unicast distribution. Coding has also been block-based rather than byte-based, resulting in longer repair latencies. The wireless standards also do not [5] consider channel coding rate adaptation.

To achieve adaptation to wireless channel conditions, channel estimation is necessary. As an example, the IEEE 802.16e standard specifies that a mobile station should provide channel measurements, which can either be Received Signal Strength Indicators or may be Carrier-to-Noise-and-Interference Ratio measurements made over modulated carrier preambles. Because Raptor coding is indeed rateless, it is possible to adaptively vary the amount of redundant data according to an estimate of the channel conditions. This estimate is then expressed as a probability of byte loss, *BL*. It is also possible to piggyback additionally generated redundant data if the estimate is insufficient to allow reconstruction of the packet or the belief propagation decoding algorithm fails to run to completion.

Apart from selection of the QP to ensure smaller packet sizes for high priority partitions, attention should be given to the Group of Pictures (GoP) structure; and the configuration of the data-partitioning in order to avoid unnecessary dependencies between partitions. Periodic intra-refresh using periodic I-frames is often avoided in wireless streaming to avoid sudden data rate increases over a constrained bandwidth. However, if intra-refresh (IR) macroblocks (MBs) are dispersed across the video sequence, the percentage of such MBs should also be considered. For example, too low a

percentage may not halt spatio-temporal error propagation. On the other hand, too high a percentage can needlessly increase the bitrate. One of the purposes of this paper is to examine how best to configure the adaptive protection scheme.

The remainder of this paper is organized as follows. Section II of the paper outlines the adaptive channel coding scheme. Section III describes the simulation model of the proposed scheme and Section IV presents an evaluation of the scheme. Finally, Section V offers some concluding remarks.

II. PROPOSED SCHEME

In the proposed adaptive scheme, the probability of channel byte loss through fast fading (*BL*) serves to predict the amount of redundant data to be added to the payload. The instantaneous *BL* is used to calculate the amount of redundant data adaptively added to the payload. In an implementation, *BL*, is found through measurement of channel conditions (refer to Section I). If the original packet length is *L*, then the redundant data are given simply by

$$\begin{aligned} R &= L\times BL+(L\times BL^2)+(L\times BL^3)\ldots \\ &= L/(1-BL) - L, \end{aligned} \tag{1}$$

which adds successively smaller additions of redundant data, based on taking the previous amount of redundant data multiplied by *BL*.

Rateless code decoding in traditional form operates by a belief-propagation algorithm [7] which is reliant upon the identification of clean symbols. This latter function is performed by PHYsical-layer forward error correction, which passes up correctly received blocks of data (through a cyclic redundancy check) but suppresses erroneous data. For example, in IEEE 802.16e, a binary, non-recursive convolutional encoder with a constraint length of 7 and a native rate of 1/2 operates at the physical layer.

In general, encoding of rateless codes is accomplished as follows: Choose d_i randomly from some distribution of degrees, where $\rho_{di} = Pr$[degree d_i]; *Pr* is the probability of a given event. Choose d_i random information symbols R_i among the k information symbols. These R_i symbols are then XORed together to produce a new composite symbol, which forms one symbol of the transmitted packet. Thus, if the symbols are bytes then all of the R_i byte's bits are XORed 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 reduces 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.

In the decoding process, the robust Soliton distribution 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

$$S = c\ln\left(\frac{k}{\delta}\right)\sqrt{k} \tag{2}$$

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

$$\begin{aligned} \tau(d) &= \frac{S}{k}\frac{1}{d} && \text{for } d = 1, 2, \ldots (k/S)\text{-}1 \\ &= \frac{S}{k}\ln\left(\frac{S}{\delta}\right) && \text{for } d = k/S \\ &= 0 && \text{for } d > k/S \end{aligned} \tag{3}$$

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

$$\mu(d) = \frac{\rho(d) + \tau(d)}{z} \tag{4}$$

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

$$z = \sum_d (\rho(d) + \tau(d)) \tag{5}$$

The following statistical model [8] can be used to model the Raptor decoding process:

$$\begin{aligned} P_f(m,k) &= 1 && \text{if } m < k \\ &= 0.85 \times 0.567^{\text{m-k}} && \text{if } m \geq k \end{aligned} \tag{6}$$

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 [8] remark and show that for $k > 200$ the model almost perfectly models the performance of the code, which implies that if blocks are used approximately 200 blocks should be received before reasonable behavior takes place. This observation motivated our 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 (6) for $k > 200$.

If a packet cannot be decoded, despite the provision of redundant data then extra redundant data are added to the next packet. In Fig. 1, packet X is corrupted to such an extent that it cannot be immediately decoded. Therefore, in packet X+1 some extra redundant data are included, up to the level that the original packet's decode-failure is no longer certain. If

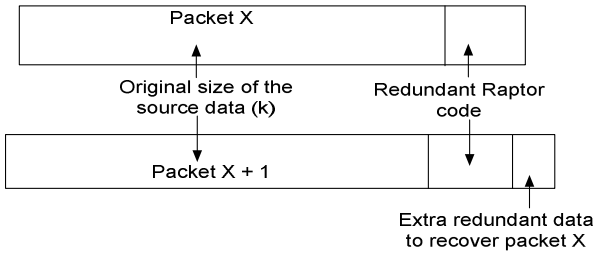

Figure 1. Division of payload data in a packet between source data, original redundant data and piggybacked data for a previous erroneous packet

according to (6), $m < k$ then a packet is corrupted to such an extent that it is effectively dropped. However, this was found in simulations to be a rare event.

It is implied from (6) that if less than k symbols (bytes) in the payload are successfully received then a further k - m + e redundant bytes can be sent to reduce the risk of failure. This reduced risk arises because of the exponential decay of the risk that is evident from equation (6) and which gives rise to Raptor code's low error probability floor.

III. SIMULATION MODEL

In the proposed adaptive scheme, the probability of channel byte loss (BL) through fast fading in the Gilbert-Elliott channel model (see later in this Section) serves to predict the amount of redundant data to be added to the payload. There are two hidden states (good and bad) in this two-state Markov model. If PGB and PBG are the probabilities of going from good to bad state and from going from bad to good state respectively, then

$$\pi_G = P_{BG}/(P_{BG} + P_{GB}) \quad (7)$$

$$\pi_B = P_{GB}/(P_{BG} + P_{GB}) \quad (8)$$

are the steady state probabilities of being in the good and bad states. PG and PB are the probabilities of (byte) loss in the good and bad states respectively. Both states were modelled by a Uniform distribution. Consequently, the mean probability of channel byte loss is given by

$$BL_{mean} = P_G . \pi_G + P_B . \pi_B \quad (9)$$

which is the mean of a Uniform distribution. Thus, in simulations BL was selected from a Uniform distribution with mean BLmean.

Assuming perfect channel knowledge of the channel conditions when the original packet was transmitted establishes an upper bound, beyond which the performance of the adaptive scheme cannot improve. However, we have included measurement noise into the estimate of BL to test the robustness of the scheme. Measurement noise was modelled as a zero-mean Gaussian (normal) distribution and added up to a given percentage (5% in the evaluation) to the packet loss probability estimate.

In simulations, the decision on whether the Raptor code belief propagation algorithm would run to completion was taken by comparing a Uniformly-distributed random variable's value with that of the probability given by (6). In the tests, e was set to four, resulting in a risk of failure (from (6)) of 8.7 % in reconstructing the original packet, if the extra redundant data successfully arrives.

The tests were performed on the reference *Paris* (moderate spatial coding complexity) and *Football* (high temporal coding complexity) video sequences encoded in Common Intermediate Format (CIF) @ 30 Hz. The GoP structure was IPPP…, i.e. one initial I-picture and all P, or IBBPBBP…, i.e. insertion of bi-predictive B-pictures for greater coding efficiency. By default, 2% intra-refresh (IR) MBs were randomly inserted. It is possible that gradual decoder refresh can be achieved by other IR MB arrangements. By default also, Constrained Intra Prediction (CIP) was set to make partition-B independent of partition-C. Notice that in the present profile structure of H264/AVC it is not possible to make partition-C completely independent of partition-B. However, partition-A is always independent.

The video stream was transmitted (refer to Fig. 2 for an example scenario) from a WiMAX base station to a mobile station (MS). To introduce sources of traffic congestion, a permanently available FTP source was configured with TCP transport to a second MS. Likewise, a CBR source with packet sizes of 1000 B (the WiMAX maximum transfer unit) and inter-packet gap of 30 ms was also downloaded to a third MS. The simulations adopted the mandatory settings for a 10.67 Mbps downlink (DL) rate with: 3:1 DL/UL sub-frame ratio for the only WiMAX forum frame size of 5 ms; and 16-QAM ½ modulation over a 10 MHz channel with IEEE 802.16e recommended antenna heights and transmit/receive powers.

The Gilbert-Elliott channel model, though it is simple, has been widely adopted [9], as it is thought to realistically model the error bursts that are apparent to the application layer. More significantly, error bursts can be particularly damaging to compressed video streams, because of the predictive nature of source coding. Therefore, the impact of error bursts should be assessed [10] in video streaming applications. In the Gilbert-Elliott model for the well-known ns-2 simulator, the settings for fast fading were PGG (probability of being in a good state) was set to 0.95, PBB (probability of being in a bad state) = 0.96, PG = 0.02 and PB = 0.165.

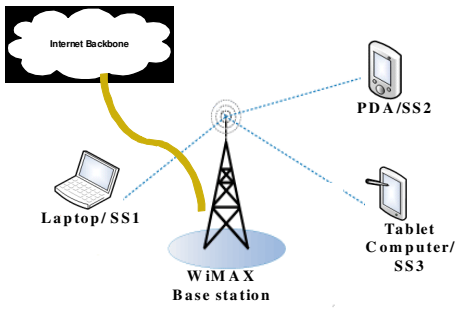

Figure 2. Example usage scenario

IV. FINDINGS

Fig. 3 is a comparison between the relative sizes of the partitions according to QP for the *Football* sequence. From the Figure, it is apparent that the partition-B's contribution increases as the percentage of IR MBs grows, making partition-B packets more vulnerable to congestion and channel error as a result. The size of partition-C bearing packets declines with increasing QP, as a result of coarser quantization of transform coefficients. Further experiments determined that 2% IR MBs in the WiMAX channel conditions achieved better video quality (PSNR) than 5%, while 25% IR MB, which is the approximate contribution of a single IR MB row, fared badly due to larger packets.

Table I shows that reducing the quality (increasing the QP towards the H.264/AVC maximum of 50) further increases the relative size of partition-A against that of partition-C. This reduction in quality causes partition-A to become relatively more vulnerable to loss. More temporally complex clips such as *Stefan* have larger partition-Cs, which are at risk of loss.

Table II shows one potential impact of the GoP structure on the scheme. B-pictures increase coding efficiency at a cost in coding complexity. However, with the inclusion of B-pictures, the mean size of P-pictures increases, as a result of the increased reference distance. For example, the IBBPBBP… mean P-picture size for QP=20 is around 15 kB.

The effect of the latter is evident in Fig. 4, in which many more packets are dropped for Football at QP=20. Dropped packets may result buffer overflow or outright channel drops (m<k in (6)). From Fig. 4, GoP structure impacts the more temporally complex sequence. The general decay in packet drop numbers in Fig. 4 results in an increase in video quality in Fig. 5. An IPPP… GoP structure is preferable except at QP = 35. However, at QP = 35 all GoP structures result in a PSNR over 25 dB, which is a boundary for 'fair' quality, when objective quality is very approximately converted to ITU mean opinion score [11].

Corrupted packet levels are generally high, Fig. 6, but in respect to GoP structure it seems that an IPPP… structure may be more favorable to temporally complex content (Football) and vice versa for less active sequences. The main consequence of higher levels of packet corruption, after the application of the proposed adaptive scheme, is in greater delay for a greater percentage of packets, Fig. 7. Apart from the anomaly at QP = 20, corrupted packet delay is approximately twice that of normal packet end-to-end delay, Fig. 8, reflecting the single retransmission of extra redundant that is permitted. It was also confirmed that for a moderate increase in mean packet size, making partition-B completely independent of partition-C through CIP (see Section III) resulted in a small (a few dB) improvement in video quality, whenever the QP setting allowed sufficient packets to be delivered.

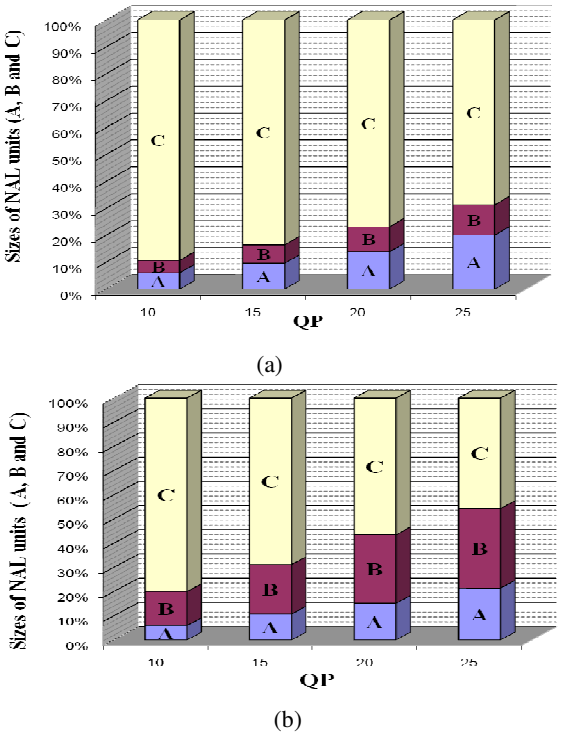

Figure 3. Relative sizes of data partitions according to quantization parameter (QP) for the Football video sequence, with (a) 5% intra-coded refresh MBs, and (b) 25% intra-coded refresh MBs

TABLE I. RELATIVE SIZES OF PARTITIONS A, B, AND C FOR VIDEO SEQUENCES PARIS AND STEFAN ACCORDING TO VIDEO QUALITY

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

TABLE II. MEAN P-PICTURE SIZE (BYTES) ACCORDING TO QP FOR DIFFERING GOP STRUCTURES.

	Football		*Paris*	
QP	IPPP…	IBBP…	IPPP…	IBBP…
20	8905	15520	5102	7590
25	6301	10311	2824	4249
30	4185	6381	1398	2238
35	2444	3756	647	1100

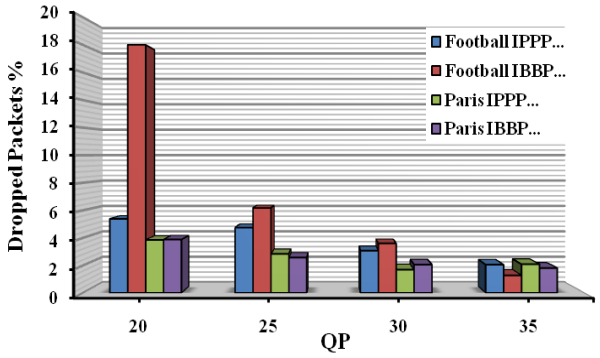

Figure 4. Dropped packets for differing GoP structure and content

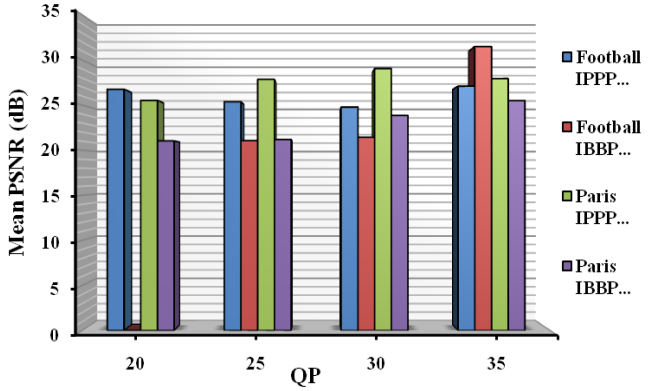

Figure 5. Video quality (Y-PSNR) for differing GoP structure and content.

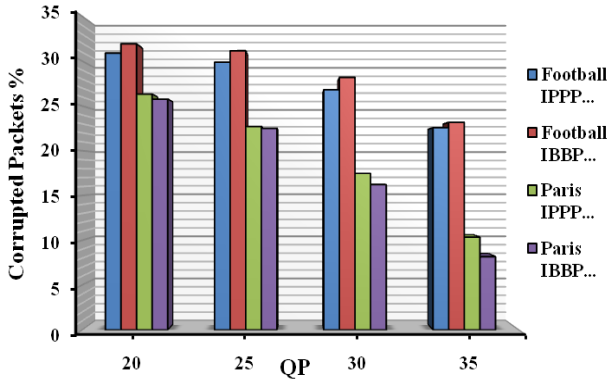

Figure 6. Percentage corrupted packets for differing GoP structure and content

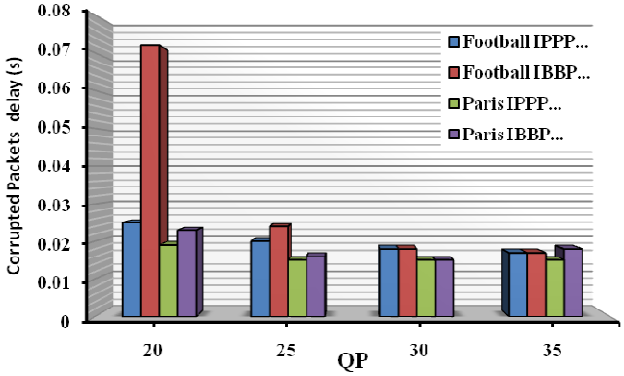

Figure 7. Mean end-to-end corrupted packet delay for differing GoP structure and content

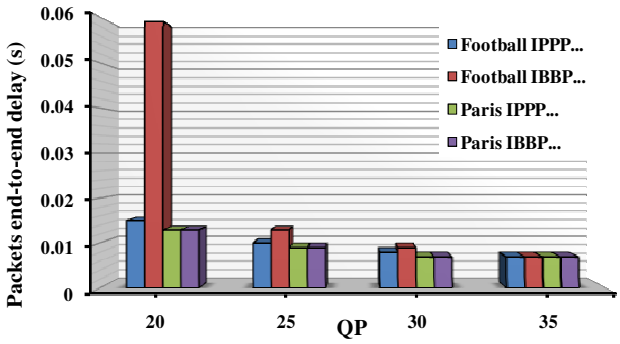

Figure 8. Mean end-to-end non-corrupted packet delay for differing GoP structure and content

V. Conclusion

This paper has presented an adaptive Raptor coded video streaming scheme. The scheme is relatively simple to implement, as no complex optimizations are involved. Consequently, it can easily be applied to mobile video streaming applications. Potential applications are user-to-user streaming, live streaming and possibly conversational applications such as video phone. The main current impediment to its implementation is that data-partitioning in an H.264/AVC codec is restricted to the Extended profile. The paper has demonstrated that to implement such a scheme the group-of-pictures structure and the quantization parameter should be carefully chosen. However, the results in this paper demonstrate suitable choices. Future work can evaluate the scheme against comparable streaming methods.

References

[1] I.V. Uilecan, C. Zhou, and G.E. Atkin, "Framework for delivering IPTV services over WiMAX wireless networks," in *IEEE Int'l Conf. on Elec./Info. Technol.*, pp. 470–475, 2007.

[2] R. Jain, C. So-In, and A.-K Al-Tamimi, "System-level modeling of IEEE 802.16e mobile WiMAX networks," *IEEE Wireless Commun.*, vol. 15, no. 5, pp. 73–79, 2008.

[3] T. Stockhammer, and M. Bystrom, "H.264/AVC data partitioning for mobile video communication," in *IEEE Int'l Conf. on Image Processing*, 2004, pp. 545-548.

[4] A. Shokorallahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.

[5] M. Luby, T. Stockhammer, and M. Watson, "Application layer FEC in IPTV services," *IEEE Commun. Mag.*, vol. 46, no. 5, pp. 95-101, 2008.

[6] E. Hepsaydir, E. Witvoet, N. Binucci, and S. Jadhav, "Enhanced MBS in UMTS networks and Raptor codes," in *IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Communs.*, 2007, pp. 1-5.

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

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

[9] 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 Trans. Circ. Systems Video Technol.*, vol. 18, no. 7, pp. 861-874, 2008.

[10] G. Muntean, P. Perry, and L. Murphy, "A new adaptive multimedia streaming system for all-IP multi-service networks," *IEEE Trans. Broadcast.*, vol. 50, no.1, pp. 1-10, 2004.