Robust IPTV Delivery with Adaptive Rateless Coding over a Mobile WiMAX Channel

Laith Al-Jobouri, Martin Fleury, and Mohammed Ghanbari

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

ABSTRACT—As intelligent content management of IPTV moves popular material nearer to the end-user, application-layer channel coding schemes, involving retransmission of extra redundant data, become attractive as a result of the reduced latency. Application-layer, adaptive rateless channel coding is exploited in this paper's scheme to reconstruct streamed video across an IEEE 802.16e (mobile WiMAX) channel. The paper concentrates on the trade-offs in implementing the scheme, showing that exact calculation of the redundant data has the potential to reduce the forward error correction bit-rate overhead. To reduce delay, an appropriate compression rate should also be selected.

I. INTRODUCTION

The BBC's iPlayer in the UK [1] has demonstrated the demand for Internet Protocol TV (IPTV) value-added video streaming in the form of content-on-demand and time-shifted TV. However, this service is primarily aimed at Asymmetric Digital Subscriber Line (ADSL) receivers and may be ill-adapted to mobile wireless broadband delivery. In such broadband wireless networks, including IEEE 802.16e (mobile WiMAX) [2], adverse channel conditions are a concern for video streaming and will become more so as the transition to higher data-rate IEEE 802.16m (WiMAX 2) [3] occurs. The 802.16e standard provides Turbo coding and hybrid ARQ at the PHYsical layer with scalable transmission bursts depending on RF conditions. However, application-layer forward error correction (AL-FEC) [4] is still recommended for IPTV during severe error conditions. This paper demonstrates packet-by-packet adaptive rateless channel coding to guard against burst errors possibly caused by slow and fast fading on a WiMAX channel.

The prior IPTV Content Delivery Network (CDN) scheme discussed in [5] was end-to-end, providing adaptation through a form of FEC simulcast. For severe conditions, it relies on the lower overhead and linear decoding complexity that one form of rateless coding, Raptor codes [6], provides. However, it now seems likely [7] that intelligent content management will result in local caching of frequently-requested content. This development enables packet-by-packet adaptive rateless coding, depending on local measurements of channel conditions. Given the reduced latency, it may also be possible to include limited retransmission of extra redundant data, made feasible through rateless channel coding.

The iPlayer, as mentioned above, is a simple simulcast service with H.264/AVC (Advanced Video Coding) codec rates [8] available at 500 kbps, 800 kbps, and 1500 kbps, which once selected are fixed to the capacity of the access network. Other related schemes (though not IPTV) such as Google's Video Talk similarly keep the compression ratio fixed but alter the screen resolution on request by the user. As the iPlayer depends on Adobe Flash Player technology, files are delivered by TCP transport, as this protocol underlies HTTP. TCP is unsuitable for real-time services over wireless because of misinterpretation of channel packet losses as packet drops through congestion. The delays introduced may well be compounded by the progressive download employed by Adobe Flash Player, which, according to [9], when used for YouTube clips may result in the cancellation of up to 10% of downloads. The need to reduce start-up delay may also lead to reductions in quality, as the initial download block must be compressed to a suitable size.

Rateless coding has been utilised [10] for 3GPP's Release 6 Multimedia Broadcast/Multicast Service (MBMS). Unfortunately, there is no direct feedback control channel [11] in an UMTS wireless access network and, hence, temporal scalability is employed in [10] in conjunction with Raptor coding without packet-by-packet adaptivity. Notice that in the current paper data-partitioned source-coding [12] is employed as a means of graceful degradation according to channel conditions. Detailed discussion of data partitioning is postponed to Section II. Another approach [13] is to use a scalable variety of rateless coding to provide unequal protection of data-partitioned video data. Growth codes can superimpose additional redundant data for the more important partition-A and -B data. However, that rateless coding scheme [13] is not adaptive either. The work in [14] explored the possibility of multiple sources generating Raptor code independently of each other to protect layers within scalable video coding. The paper investigated the coordination of the scheme to achieve rate-distortion optimization. In fact, perhaps the nearest scheme to ours provided for Internet video streaming [15]. However, it is for single-layer delivery and accepts long latencies.

In a mobile WiMAX channel, the packet size critically affects the risk of packet corruption. If it is possible to estimate the channel conditions then the amount of redundant data can be set accordingly, thus controlling the packet size. If H.264/AVC data-partitioning [16] is employed and if the quantization parameter is selected appropriately then the packet size decreases as the compressed data priority increases. Higher-priority packets are sufficient to partially reconstruct a video frame. Reducing the size of the redundant component of a packet rather than employing a fixed ratio of redundant data becomes particularly appropriate in this type of streaming. The WiMAX standard already specifies that a station 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.

In a realistic adaptive scheme, perfect channel knowledge cannot be assumed, as a channel estimate will be affected by measurement noise. If it is not possible to reconstruct a packet with the amount of redundant data available then in

the proposed scheme a single Automatic Repeat request (ARQ) is permitted (to avoid excessive delay to the video-rate application by more ARQs), again allowing the rateless features of the channel code to be exploited. There is, however, a danger that if the number of packets corrupted during transmission increases then the overall delay will increase significantly. Two variants of an adaptive rateless scheme are introduced in this paper. In the first, additional redundant data is adjusted up to the amount needed to prevent retransmission of redundant data (assuming perfect channel knowledge). This comes at a cost in increased overhead but reduces overall delay. The second variant dynamically calculates the amount of adaptive redundant data required to match the probability of error. Though it reduces the FEC overhead it may introduce extra delay. However, delay can also be adjusted by varying the video quality and, hence, the packet sizes.

If due to traffic congestion buffer overflow occurs, packets will be dropped outright. Should this occur then rateless coding is powerless to protect. However, packet duplication may be applied. Without data-partitioning it is necessary to completely duplicate slices. However, with data-partitioning it is possible to only duplicate part of the bit-stream, namely those packets carrying the most important packets (header and motion information). Though the main theme of the paper is the role of the adaptive scheme to protect against packet corruption, protection of packets through duplication is also assessed.

The Section II introduces background material necessary for a fuller understanding of this paper. Section III then describes the adaptive rateless scheme. The Section IV describes the experimental methodology employed to derive the results evaluated in Section V. Section VI concludes this paper.

II. BACKGROUND

This Section principally introduces rateless channel coding and data-partitioning. Some results are also given to explain the interest in data-partitioning.

A. Rateless Codes

Rateless or Fountain coding [17], of which Raptor coding [6] is a subset, is ideally suited to a binary erasure channel in which either the error-correcting code works or the channel decoder fails and reports that it has failed. In erasure coding, all is not lost as flawed data symbols may be reconstructed from a set of successfully received symbols (if sufficient of these symbols are successfully received). A fixed-rate (n, k) 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, in practice not only must n, k, and q be small but also the computational complexity of the decoder is of order $n(n - k) \log_2 n$. The erasure rate must also be estimated in advance.

The class of Fountain codes [17] 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 [18] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure. The 'belief propagation' decoding relies on the column entries of the generator matrix being selected from a robust Soliton distribution. In the LT generator matrix case, the expected number of degree one combinations (no XORing of symbols) is $S = c \ln(k/\partial)\sqrt{k}$, for small constant c. (see later discussion). Setting $\varepsilon = 2 \ln(S/\partial) S$ ensures that by sending $k(1 + \varepsilon)$ symbols these symbols are decoded with probability $(1 - \partial)$ and decoding complexity of order $k \ln k$.

Encoding of the LT in the form used in this paper is accomplished as follows: Choose d_i randomly from some distribution of degrees, where $\rho_{di} = Pr[\text{degree } d_i]$, Pr is the probability 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, as mentioned above.

Symbols are processed at the decoder as follows. If a symbol arrives with degree greater than one it is buffered. If a clean symbol arrives with degree one then it is XORed with all symbols in which it was used in the encoding process. This reduces the degree of each of the symbols to which the degree one symbol is applied. When a degree two symbol is eventually reduced to degree one, it too can be used in the decoding process. Notice that a degree one symbol for which no *XOR*ing has taken place. Notice also that for packet erasure channels a clean degree one symbol (a packet) is easily established as such.

The ideal Soliton distribution [18] is a first stage in finding an appropriate degree distribution. It is defined as:

$$\rho(1) = 1/n \tag{1}$$

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

where k is the number of source symbols.

As already mentioned, in practice, the robust Soliton distribution [18] is employed as this produces degree-one symbols at a more convenient rate for decoding. It also avoids isolated symbols that are not used elsewhere. Two tuneable

parameters c and δ are used to form the expected number of useable degree one symbols. We have already given the following equation in the prior discussion:

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

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

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

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

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

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

$$z = \sum_d (\rho(d) + \tau(d)) \qquad \cdot \tag{6}$$

The essential differences between Fountain erasure codes and RS erasure codes are that: Fountain codes in general (not Raptor codes [6]) 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. Apart from the startling reduction in computational complexity, a Raptor code has the maximum distance separable property, that is, the source packets can be reconstructed with high probability from any set of k or just slightly more than k received symbols. A further advantage of Raptor coding is that it does not share the high error floors on a binary erasure channel [10] of prior rateless codes.

B.   Data partitioning

In an H.264/AVC codec [8], the Network Abstraction Layer (NAL) facilitates the delivery of the Video Coding Layer (VCL) data to the underlying protocols such as RTP/UDP/IP, H.32X and MPEG-2 transportation systems. Compressed data are grouped by the codec into self-contained sub-video frame units called slices. Normally a slice is encapsulated by the codec into a NAL unit (NALU)). Each NALU can be considered as a packet that contains an integer number of bytes including a header and a payload. The header specifies the NALU type and the payload contains the related data. When data partitioning is enabled, every slice is further divided into three separate partitions and each partition is located in either of type 2 to type-4 NALUs, as listed in Table I. A NALU of type 2, also known as partition-A, comprises the most important information of the compressed video bit stream of P- and B-pictures, that is the MB addresses, motion vectors (MV's) and essential headers. If any MBs in these pictures are intra-coded, their transform coefficients are packed into a type-3 NALU, also known as partition-B. Type 4 NAL, also known as partition-C,

carries the transform coefficients of the motion-compensated inter-picture coded MBs. Partitions-A and -B of data-partitioned P- and B-slices are small for broadcast quality video but their C-type partitions can be very long.

TABLE I.   NALU 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

Data partitioning is a form of source-coded error resilience [20]. Combining, error resilience with error control involves additional data overhead. However, Fig. 1 shows that, of four common error resilience tools in H.264/AVC, data partitioning has the least overhead. The illustration is the well-known *Foreman* clip representing the jerky motion of a hand-held camera with a rapid pan towards the end of the sequence. In Fig. 1, the horizontal axis represents the mean bitstream rate arrived at by setting the QP to the given value, while the vertical axis represents the mean overhead rate with that QP. As the quality decreases (higher QP) the advantage of data-partitioning increases as the relative overhead of all schemes increases. Tests of the *Akiyo*, *Coastguard*, and *Mobile* sequences show that the overhead is not strongly dependent on coding complexity, with the size of overhead ordering between the schemes preserved.
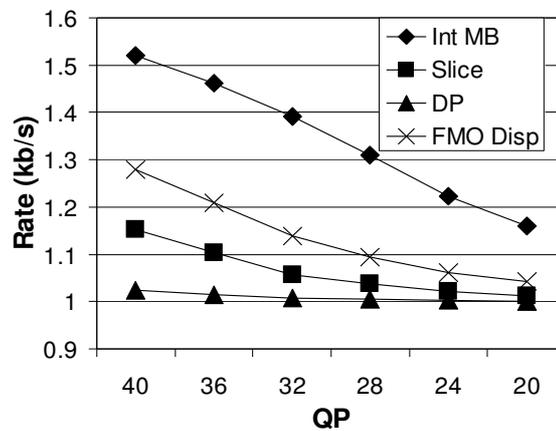

Figure 1 - QCIF *Foreman* rates at 15 frame/s with 4:2:0 chroma sub-sampling according to QP (horizontal axis) plotted against overhead rate (vertical axis) arising from H.264 error resilience tool: Int MB = Intra-coded macroblock refresh, FMO Disp = Flexible Macroblock Ordering dispersed mode (two slices), DP = data-partitioning, Slice = slice structuring with 3 slices per frame.

The relative mean sizes (across all frames in the sequence) of the data partitions for a sequence with higher spatial coding complexity, *Paris,* and one with high temporal coding complexity, *Stefan,* were examined. The results for these sequences are reported in Table II according to video quality given by the QP setting. Both sequences with 4:2:0

sampling were Variable Bitrate (VBR) encoded at Common Intermediate Format (CIF) (352×288 pixel/frame), with a Group of Picture (GoP) structure of IPPP….. at 30 frame/s. Experiments not shown indicate that including B-pictures, with a GoP structure of IPBP... (sending order) and an intra-refresh rate of 15, did not noticeably change the distribution of partition sizes.

TABLE II.   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%

C.   Intra-refresh macroblocks

The insertion of intra-refresh (IR) MBs into pictures [21] normally encoded through motion-compensated prediction allows temporal error propagation to be arrested if matching MBs in a previous picture are lost. In the H.264/AVC JM implementation, various IR schemes exist such as random, which sets a maximum percentage of MBs, or cyclic, which replaces each line of the picture in turn in cyclic order. Notice that naturally encoded IR MBs are also inserted into predictively-coded P-pictures when inter-coding brings limited or no advantage. For example, this may occur during rapid motion or when a new object that is not present in a previous picture is uncovered. The inclusion of IR MBs does lead to some increase in the size of partition-B bearing packets as shown in Table III for different QPs and percentages of IR MBs. The sequence is *Football* with high temporal coding complexity, encoded with the same configuration as in Section II.B. It is also possible to vary the IR rate according to scene type or channel conditions [22].

TABLE III.   TOTAL NALU SIZES IN BYTES FOR DIFFERENT IR MB PERCENTAGES AND MB LINE INTRA UPDATE FOR CIF PICTURES IN THE *FOOTBALL* SEQUENCE

QP	2% Intra-refresh MB			5% Intra-refresh MB		
	A	B	C	A	B	C
20	1842	2678	3889	1845	2767	3867
25	1687	1697	2533	1690	1763	2511
30	1459	1047	1496	1463	1082	1482
35	1117	572	688	1120	595	682
QP	25% Intra-refresh MB			MB Line Intra Update		
	A	B	C	A	B	C
20	1893	3450	3669	1885	3385	3683
25	1746	2216	2379	3683	2160	2400
30	1505	1346	1405	1498	1312	1414
35	1146	729	646	1143	716	652

D. Slice duplication

In the extended protection scheme, one of the data-partitions is duplicated so that dropped packets can be replaced. Rather than duplicate or copy, it is possible to send reduced-quality versions of a slice, which in H.264/AVC is called

a redundant slice [23]. However, employing a coarser quantization than the main stream, can lead to drift between the encoder and decoder, as the encoder never knows which version of the slice has been decoded. Besides, replacing one partition with a redundant slice with a different QP to the other partitions would not even permit reconstruction.

A possibility [23] is to use correctly-received reference pictures for reconstruction of redundant pictures rather than the reference pictures used by primary pictures. The decoder is able to select from a set of potential replacement redundant pictures according to the possibility of correct reconstruction. Alternatively, in [24], MBs were selected for their relative impact on reconstruction and placed within FMO slices, at some increase in computational complexity.

III. ADAPTIVE SCHEME

In the adaptive scheme, the probability of channel loss (*PL*) serves to predict the amount of redundant data to be added to the payload. Assume that 'bursty' error conditions are generated by the widely-used Gilbert-Elliott model [25] [26], which is a form of hidden Markov model with a good and bad state. In the Gilbert-Elliott model, Fig. 2, p_{gb} is probability of transition from the good state to the bad state and p_{bg} is the probability of transition from the bad state to the good state. Then p_{gg} and p_{bb} are the probabilities of staying in the good state and the bad state respectively:

$$p_{gg} = 1 - p_{gb}, p_{bb} = 1 - p_{bg} \tag{7}$$

Steady-state probabilities for staying in the good and bad state are defined as π_G and π_B and are found to be.

$$\pi_G = \frac{p_{bg}}{p_{bg} + p_{gb}}, \pi_B = \frac{p_{gb}}{p_{bg} + p_{gb}} \tag{8}$$

Consequently, the mean probability of loss is given by

$$PL_{mean} = PG. \pi_G + PB. \pi_B \tag{9}$$

where *PG* and *PB* are the probabilities of loss in the good and bad states respectively. The instantaneous *PL* (taken from a distribution with mean PL_{mean}) is used to calculate the amount of redundant data adaptively added to the payload.
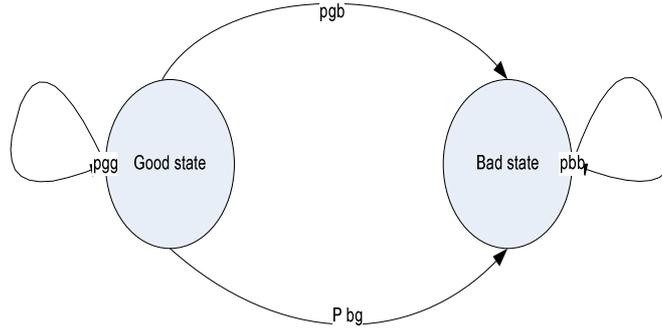
Figure 2 - Gilbert-Elliott two-state model showing transition probabilities

If the original packet length is L, then the redundant data is given simply by

$$R = L \times PL \times A \tag{10}$$

However, the factor A must accommodate all values of PL for a particular value of PB. Subsequent tests reported in Section IV showed that factor A can be dispensed with in favor of a dynamically determined value for the redundant data:

$$R = L \times PL + (L \times PL^2) + (L \times PL^3)\ldots$$
$$= L/(1-PL) \tag{11}$$

with successively smaller additions of redundant data, based on taking the previous amount and multiplying by PL.

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 to test the robustness of the scheme. Measurement noise was modelled as a zero-mean (truncated) Gaussian (normal) distribution and added to the packet loss probability estimate.

If despite the redundant data the packet's payload still cannot be reconstructed then extra redundant data are piggybacked onto the next packet. For example, in order to model Raptor coding, the following statistical model [27] was employed:

$$P_f(m,k) = 1 \qquad \text{if } m < k,$$
$$= 0.85 \times 0.567^{m-k} \qquad \text{if } m \geq k, \tag{12}$$

where $P_f(m,k)$ is the failure probability of the code with k source symbols if m symbols have been received. Notice that the authors of [27] remark and show that for $k > 200$ the model almost perfectly models the performance of the code.

As previously remarked, the rateless channel code symbol size was set to a byte. Clearly, if the symbol size is a packet and instead 200 packets are accumulated before the rateless decoder can be applied (or at least equation (12) is relevant) there is a penalty in start-up delay for the video stream and a cost in providing sufficient buffering at the mobile stations.

IV. METHODOLOGY

This Section presents technical details of the evaluation methodology.

A. Detecting errors

Rateless code decoding is reliant upon the identification of clean symbols. This latter function is performed by PHYsical-layer FEC which passes up correctly received blocks of data (through a Cyclic Redundancy Check) but suppresses erroneous data. For example, in IEEE 802.16e [1], a binary, non-recursive convolutional encoder with a constraint length of 7 and a native rate of 1/2 operates at the physical layer. Upon receipt of the correctly received data, decoding of the information symbols is attempted, which will fail with a probability given by (1) for k > 200. This implies from (12) that if less than k clean symbols (bytes) in the payload are successfully received then *k-m+e, e > 0* redundant bytes can be sent to reduce the risk of failure. In tests, *e* = 4, resulting in an 8.7% risk of failure because of the exponential decay of the risk evident from equation (12). The extra data are additional data over and above the adaptively-estimated redundant data originally added to the packet.

To reduce the network path latency, the number of retransmissions, after an ARQ over the uplink, was limited to one. Recall that there are strict display deadlines for video decoding. Fig. 3 shows how ARQ triggered retransmissions work. In the Figure, the payload of packet X is corrupted to such an extent that it cannot be reconstructed. Therefore, in packet X+1 some extra redundant data are included up to the level that its failure is no longer certain. If the extra redundant data are insufficient to reconstruct the original packet's payload, the packet is simply dropped to avoid further delay. Otherwise, of course, the payload is passed to the H.264/AVC decoder.
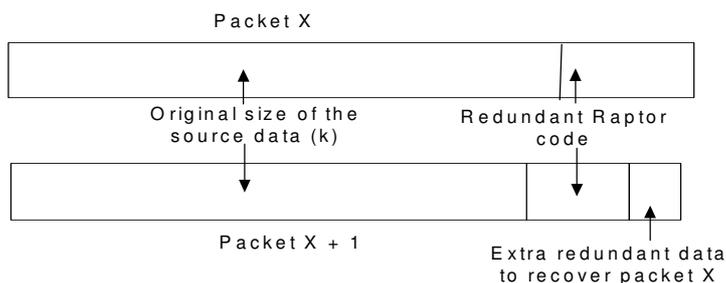

Figure 3 - Division of payload data in a packet (MPDU) between source data, original redundant data and piggybacked data for a previous packet.

In the extended protection scheme with duplication, both original and duplicate packets were protected by rateless channel coding. However, if both packets are found to be corrupted, the receiver decides to request retransmission of

piggybacked redundant data for the least corrupted packet. Piggybacking only takes place for the original stream packets.

B. WiMAX simulation configuration

To establish the behavior of rateless coding under WiMAX the ns-2 simulator augmented with a module [28] that has proved an effective way of modeling IEEE 802.16e's behavior. For the Gilbert-Elliott model parameters, p_{gg} (probability of remaining in a good state) was set to 0.95, p_{bb} (probability of remaining in a bad state) = 0.96, PG (probability of byte corruption in a bad state) = 0.02 and PB (probability of byte corruption in a bad state) was made variable taking values 0.05, 0.10, 0.15, and 0.165. These values were *not* chosen because they represented the underlying physical characteristics of a particular channel but because they represent error statistics [26] seen by an application. Burst errors can be particularly damaging to compressed video streams, because of the predictive nature of source coding. Therefore, the impact of 'bursty' errors [29] should be assessed in video streaming applications. In this case, we were interested in settings that implied high levels of packet corruption and hence the risk of significant delay to a video streaming application.

The PHY layer settings selected for WiMAX simulation are given in Table IV. The antenna heights and transmit power levels are typical ones taken from the Standard [30]. 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. Similarly, multiple-input multiple-output (MIMO) antennas are not modeled. The IEEE 802.16e Time Division Duplex (TDD) frame length was set to 5 ms, as only this value is supported [31] in the WiMAX forum adaptation of the Standard. The data rate results from the use of one of the mandatory coding modes [30] for a TDD downlink/uplink sub-frame ratio of 3:1. The BS was assigned more bandwidth capacity than the uplink to allow the WiMAX BS to respond if necessary to multiple mobile devices. Thus, the parameter settings in Table IV such as the modulation type and PHY coding rate are required to achieve a datarate of 10.67 Mbps over the downlink.

TABLE IV. IEEE 802.16 PARAMETER SETTINGS

Parameter	Value
PHY	OFDMA
Frequency band	5 GHz
Bandwidth capacity	10 MHz
Duplexing mode	TDD
Frame length	5 ms
Max. packet length	1024 B
Raw data rate	10.67 Mbps
Downlink/Uplink ratio	3:1
IFFT size	1024
Modulation	16-QAM 1/2
Guard band ratio	1/8
MS/BS transmit power	245 mW, 20 W
Approx. range to MS	1 km
Antenna type	Omni-directional
Antenna gains	0 dBD
MS/BS antenna height	1.2 m, 30 m

OFDMA = Orthogonal Frequency Division Multiple Access, QAM = Quadrature Amplitude Modulation, TDD = Time Division Duplex, MS = mobile station

C. Congestion sources and packet losses

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 a second mobile station (MS). Likewise a Constant Bit-Rate (CBR) source with packet size of 1000 B and inter-packet gap of 0.03 s was also downloaded to a third MS. WiMAX supports a set of Quality-of-Service queues. While the CBR and FTP traffic occupies the WiMAX non-rtPS (non-real-time polling service) queue, rather than the rtPS queue, they still contribute to packet drops in the rtPS queue for the video, if the packet rtPS buffer is already full or nearly full, while the nrtPS queue is being serviced.

However, in the first set of tests in which VBR video was used, buffers were set to such a value that no drops through buffer overflow occurred. In the second set of tests in which CBR video was used, *PB* was increased to 0.165. In addition, packet loss through buffer overflow was now introduced as buffer sizes were set to 50 packets (a single Medium Access Control (MAC) Service Data Unit with a WiMAX MAC Protocol Data Unit). This buffer size was selected as appropriate to mobile, real-time applications for which larger buffer sizes might lead both to increased delay and larger memory energy consumption in mobile devices. As a point of comparison, capacity studies [31] suggest up to 16 mobile TV users per mobile WiMAX cell in a 'lossy' channel depending on factors such as the form of scheduling and whether MIMO is activated.

D. Video configuration

The JM 14.2 version of the H.264/Advanced Video Coding (AVC) codec software was employed with the Evalvid environment [32] used to reconstruct sequences according to reported packet loss from the simulator and assess the video quality (PSNR) relative to the input YUV raw video. The reference *Football* video sequence was employed for the WiMAX downlink tests. *Football* was VBR encoded with 4:2:0 chroma subsampling in CIF at 30 frame/s. *Football*'s rapid motion is a cause of its coding complexity, making it a difficult test of the system. A frame structure of IPPP.... was employed to avoid the data-rate increases and delay associated with periodic I-frames. With all predictive P-frames except the initial frame, it was necessary to protect against temporal error propagation in the event of P-frame slices being lost. To ensure higher-quality video, up to 5% intra-refresh macroblocks (MBs)[1] (randomly placed) were included in each frame (apart for the first I-frame) to act as anchor points in the event of slice loss.

V.   EVALUATION

This Section considers the adaptive rateless coding scheme next, followed by the extended version of the protection scheme with duplicated packets in Section V.A.

A. Adaptive scheme recovery from corrupted packets

Initial empirical investigation showed that there was insufficient provision, unless approximately 10% extra data were added over and above that allowed for by a direct use of the instantaneous value of *PL* to estimate the redundant data. Moreover, this adjustment varies according to channel conditions, though the change monotonically increases according to *PL*. For example, Table V shows some sample values of factor *A* from equation (10) found during these investigations.

There are two potential gains from applying the adaptive scheme from formula (11), rather than a fixed factor *A* according to Table V. The first is that Table V's values can only be arrived at through a considerable number of tests for a particular scenario. The second gain is that Table IV's values may over-compensate with redundant data and consequently require extra bandwidth.

Corrupted packets are those that are received but affected by Gilbert-Elliott channel noise to such an extent that they cannot be reconstructed without additional piggybacked redundant data. In general, because extra redundant data is retransmitted it is likely that most packets will be repaired. However, a rising percentage of corrupted packets will result in increased delay. This delay will affect interactive applications. In Fig. 4, the percentage of corrupted packets is recorded according to a variation in the value of QP and increasing probability of data loss in the bad state, *PB*. Recall from Section II.B that varying QP changes the ratio between the data-partitioning packet sizes as well as

---

[1] Notice that in the JM implementation the given precentage of intra-refresh macroblocks actually represents a maximum percentage of intra-coded MBs, including naturally encoded MBs.

changing the overall size of the compressed data for any one picture. The conditions for transmission (refer to Section IV.C) for this set of tests included large enough buffers to avoid buffer overflow and reduced risk of outright packet loss from channel conditions. These conditions are unlikely to be met in practice and, thus, Fig. 4's results represent an upper bound to performance. For the response when there is outright packet loss refer to Section V.B.

TABLE V.   EMPIRICAL VALUES FOR FACTOR A

PB	Factor A	PL (%)
0.05	1.036	3.6
0.10	1.070	6.4
0.15	1.099	9.2
0.20	1.138	12.0
0.25	1.171	14.7
0.30	1.220	17.5
0.35	1.260	20.0

In Fig. 4a, when $PB = 0.05$, assuming no measurement noise results in zero packet loss whether a fixed or variable factor is used in the estimation of additional data. (Zero loss is represented by a flat bar in Fig. 4a.) This is because the exact fixed factor for this PB has been selected. For the same reason, when various percentages of Gaussian measurement noise (GN) are added the percentage of corrupted packets is better when the fixed factor is used. For both schemes, when video quality is reduced then the percentage of corrupted packets also reduces, as packet lengths reduce. However, when the Gilbert-Elliott PB value is increased but the fixed factor is not changed then the percentage of corrupted packets increases, as shown for $PB = 0.10, 0.15$ in Fig. 4b and 4c. Thus, in Fig. 4b and 4c, the variable scheme is now superior.

An interesting feature of these results is that adding measurement noise to the estimate of packet loss, PL, may actually cause less corrupted packets to occur (as more redundant data may be added). Keeping the fixed factor constant in Fig. 4b and 4c represents the situation when a misestimate of this factor has occurred. From Table V it can be seen that the misestimate may only need to be by a small percentage before weaker performance results than the variable factor scheme.

Table VI shows the resulting objective video quality and the packet delay, for the packets that did not receive retransmitted data before they were passed to the decoder and those that did (corrupted packets). Though Table VI is for $PB = 0.05$, variable factor adaptive scheme with 2% additive Gaussian measurement noise, other results were very similar. Because no packets are dropped outright, the video quality is high, as it only represents compression distortion. The real impact arises from the end-to-end delay introduced by the need to retransmit extra redundant data, because the proportion of such packets can considerably prolong the streaming period. The time taken to send packets is also dependent on the packet sizes, which reflect the QP setting for the VBR video.
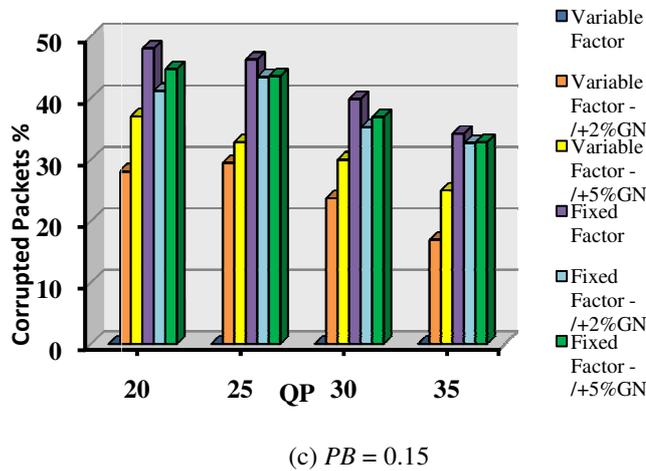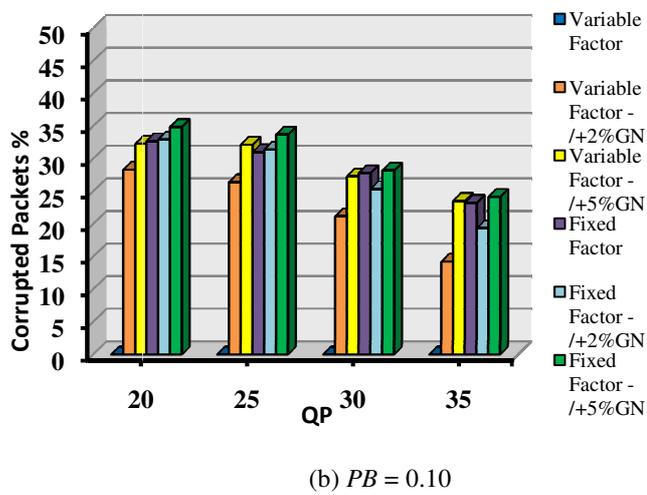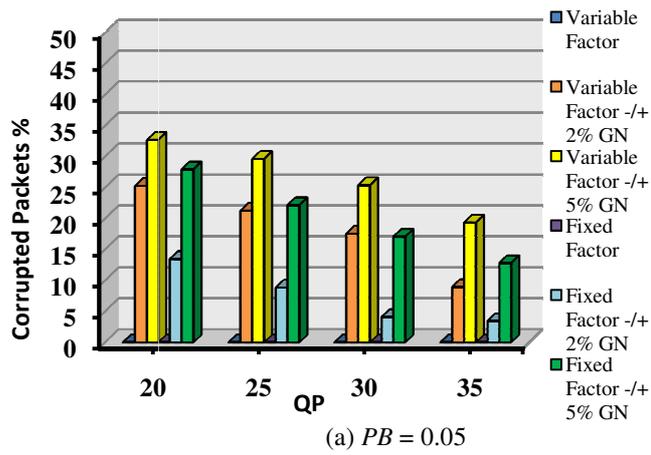
(a) *PB* = 0.05

(b) *PB* = 0.10

(c) *PB* = 0.15

Figure 4 - Variable versus fixed factor adaptive rateless coding.

TABLE VI.   VIDEO QUALITY AND DELAY FOR ADAPTIVE SCHEME

QP	PSNR (dB)	Mean packet end-to-end delay (s)	Mean corrupted packet end-to-end delay (s)
20	45.35	0.0127	0.0224
25	42.49	0.0098	0.0194
30	38.44	0.0081	0.0182
35	33.54	0.0069	0.0171

B. Adaptive scheme recovery from corrupted packets and packet drops

Though research studies often assume VBR transmission, as this represents a way to maintain consistent video quality, in practice a decision is often made by broadcasters to use CBR transmission. CBR streaming has the advantage that storage and/or bandwidth requirements can be reserved in advance. This Section tests CBR transmission at two different rates, 500 kbps and 1 Mbps. As mentioned in Section IV.C, the buffer sizes were set to a lower but realistic size introducing a risk of outright packet loss from buffer overflow.

To compensate for the risk of losing vital information in partition-A, a duplicate packet carrying partition-A was introduced in some of the tests. This is compared in Table VII for *Football* with complete duplication of a slice. For example, at 1 Mbps, duplication of partition-A adds an extra 37% to the data-rate, while duplicating a complete slice obviously adds an extra 100% to the data-rate. Because *Football* is relatively complex to code, its partition-A contribution is smaller but conversely the impact of packet corruption is higher, because of predictive dependency in the video encoding process.

TABLE VII. PERCENTAGE OF DATARATE FROM EACH PARTITION

Partition:	A	B	C
Football at 500 kbps	52%	14%	34%
Football at 1Mbps	37%	24%	39%

Fig. 5 shows the number of packets dropped outright when streaming *Football* with and without data-partitioning. In the case of duplication, a packet is lost when no duplicate is available. For data-partitioning streaming, obviously more packets are sent. Duplication in all cases reduces the packet drop rate (because missing packets can be replaced by their duplicates) and in the case of streaming at 1 Mbps without data-partitioning, the reduction is comparatively large. The gain for the scheme without data-partitioning is relatively larger, though for generally fewer and larger packets. From Fig. 6, the luminance PSNR is better with duplication than without and greater gain results from using duplication at the higher data-rate. Video quality at 25–31 dB is approximately equivalent to an ITU-T recommendation P.910 'fair' rating, whereas above 31 dB it is 'good'. Thus, duplication is needed to pass these quality thresholds. However, the main point is that data-partitioning results in equivalent objective video quality to slice duplication at a higher data rate.
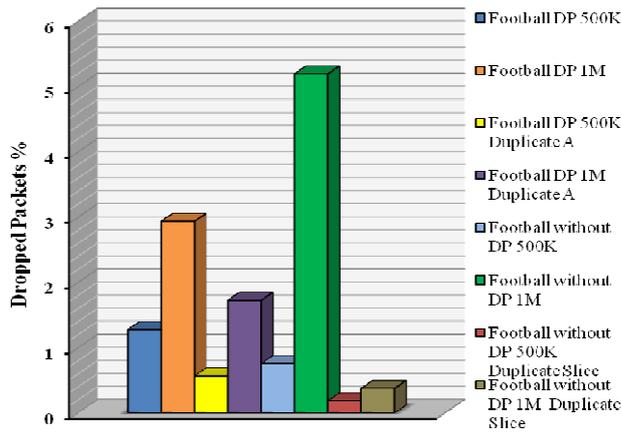
Figure 5 - Dropped packets according to CBR rate, presence of data-partitioning (DP) and/or duplicate slices. 500K = 500 kbps, 1M = 1 Mbps
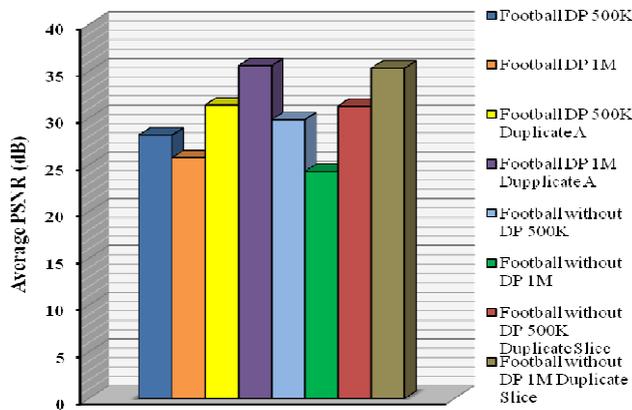

Figure 6 - Y-PSNR according to CBR rate, presence of data-partitioning (DP) and/or duplicate slices. 500K = 500 kbps, 1M = 1 Mbps.
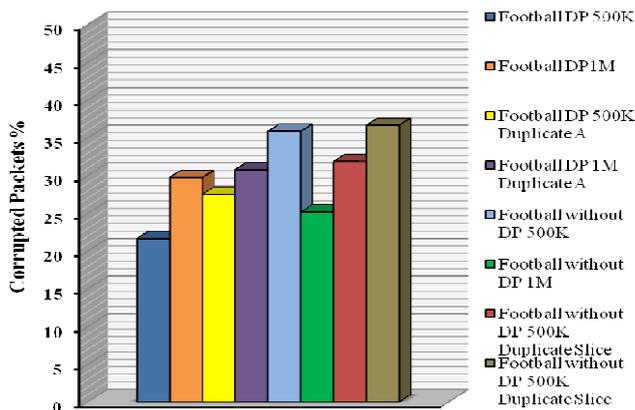

Figure 7 - Corrupted packets according to CBR rate, presence of data-partitioning (DP) and/or duplicate slices. 500K = 500 kbps, 1M = 1 Mbps.

However, this gain is only achieved with adaptive Raptor channel coding protection, as is apparent for the high levels of packet corruption in Fig. 7, as previously reported in Section V.A. In all but one case, the percentage of packets corrupted is larger for the non-data partitioning scheme though the number of packets sent is larger when employing data-partitioning. As also previously observed, the main impact of packet corruption is delay from retransmission. The non-data-partitioned scheme incurred greater delay (a mean of 4 to 5 ms for 500 kbps and 1 Mbps with duplication compared to 2 ms for the equivalent data-partitioned packets), as well as a greater risk of corruption. These delays are generally lower than streaming with VBR video, principally because of the more regular sizes of CBR packets but also because of the reduced number of packets after packet losses contributing to the mean delay. Though the delays are generally small, there is the possibility of accumulated delay leading to missed display deadlines for long video streams.

VI. CONCLUSION

Two adaptive rateless channel coding schemes were presented. To reduce the number of corrupted packets it is possible to empirically estimate the redundant data quantity that will minimize the corrupted packet number (assuming some measurement noise in estimating the packet loss rate). However, this estimate must be made before transmission begins and must be made for each possible channel condition. The paper shows that in practice a dynamically calculated redundant data overhead can be effective. The scheme will also reduce the FEC overhead. Whichever scheme reduces the number of corrupted packets reduces the overall delay introduced into the video stream. However, if packets are simply not received then the adaptive rateless coding scheme cannot help. However, by duplicating the more important partition-A packets it is shown that video quality remains acceptable with a moderate increase in the data-rate. As CDNs have brought data closer to the end user, latency is likely to be reduced. Consequently, the protection scheme presented represents a way forward for CDN video streaming.

REFERENCES

[1] "BBC iPlayer growth continues as bbc.co.uk records 29% increase," Feb. 2008 [Online] Available: http://www.bc.co.uk/pressoffice/

[2] R. Jain, C. So-In, and A.-K. Al-Tamimi, "Systems-level modeling of IEEE 802.11e mobile WiMax networks," IEEE Wireless Communs., vol. 15, no. 5, pp. 73-79, 2008.

[3] S. Ahmadi, Mobile WiMAX: A systems approach to understanding IEEE 802.16m radio access technology. Academic Press, London, UK.

[4] DVB Blue Book A115 — DVB Application Layer FEC Evaluations; http://www.dvb.org/technology/bluebooks, May 2007.

[5] T. Stockhammer, M. Luby, M. Watson, "Application layer FEC in IPTV services", IEEE Communs. Mag., vol. 46, no. 5, pp. 94-101, May 2008.

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

[7] N. Degrande, K. Laevens, and D. De Vleeschauwer, "Increasing the user perceived quality for IPTV services", IEEE Communs. Mag., vol. 46, no. 2, pp. 94-100, Feb. 2008.

[8] 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, 2003.

[9] P. Gill, M. Arlitt, Z. Li and A. Mahanti, "YouTube traffic characterization: A view from the edge", 7th ACM SIGCOMM Conf. on Internet Measurement, pp. 15–18, 2007.

[10] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall, "Delivery of broadcast services in 3G networks", IEEE Trans. Broadcasting, vol. 53, no. 1, pp. 188–199, 2007.

[11] C. Hellge, T. Schierle, J. Husch, T. Rusert, M. Kampman, and T. Wiegand, "Graceful degradation in 3GPP MBMS mobile TV services using H.264/AVC temporal scalability", EURASIP J. on Wireless Communs. and Networking, on-line journal, 15 pages, 2009.

[12] S. Mys, P. Lambert, W. de Neve, "SNR scalability in H.264/AVC using data partitioning", Pacific Rim Conf. in Multimedia, pp. 329-338, 2006.

[13] R. Rouzbeh, M. Fleury, M. Altaf, H. Sammak, and M. Ghanbari, "H.264 video streaming with data-partitioning and growth codes", Int'l Conf. on Image Processing, pp. 909-912, 2009.

[14] T. Schierl, S. Johansen, C. Hellge, T. Stockhammer, and T. Wiegand, "Distributed rate-distortion optimization for rateless coded scalable video in mobile ad hoc networks", IEEE Int'l Conf. on Image Processing, pp. 497–500, 2007.

[15] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Adaptive unicast video streaming with rateless codes and feedback", IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 2, pp. 275-285.

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

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

[18] M. Luby "LT codes", 34rd Ann. IEEE Symp. on Foundations of Computer Science, 2002, pp. 271–280.

[19] R. Palanki, and J. Yedidai, "Rateless codes on noisy channels", Int. Symp. Info. Theory, 2004, p. 37.

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

[21] R.M. Schreier, and A. Rothermel, "Motion adaptive intra refresh for low-delay video coding", Int'l Conf. on Consumer Electronics, 2006, pp. 453-454.

[22] Y.J. Liang, K. El-Maleh, and S. Manjunath, "Upfront intra-refresh decision for low-complexity wireless video telephony", Int'l Symp. on Circuits and Syst., 2006, 4 pages.

[23] C. Zhu, Y.K. Wang, M. Hannuksela, and H. Li, "Error resilient video coding using redundant pictures", IEEE Int'l Conf. on Image Processing , 2006, pp. 801–804.

[24] P. Ferré, D. Agrafiotis, and D. Bull, "A video error resilience redundant slices algorithm and its performance relative to other fixed redundancy schemes", Image Communication, vol. 25, no. 3, pp. 163-178, 2010.

[25] M. Zorzi, and R. Rao, "On the statistics of block errors in bursty channels", IEEE Trans. Communs., vol. 45, no. 6, 660–667, 1997.

[26] Ch. Jiao, L. Schwiebert, and B. Xu, "On modeling the packet error statistics in bursty channels**",** IEEE Conf. on Local Computer Networks, 534- 541, 2002.

[27] 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.

[28] F.C.D. Tsai, et al., "The design and implementation of WiMAX module for ns-2 simulator," Workshop on ns2, article no. 5, 2006.

[29] 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. Syst. Video Technol., vol. 18, no. 7, pp. 861-874, 2008.

[30] 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.

[31] C. So-In, R. Jain, and A.-K. Tamimi, "Capacity evaluation for IEEE 802.16e mobile WiMAX," *Journal of Computer Systems, Networks, and Commun.* [online], 12 pages, 2010.

[32] J. Klaue, B. Rathke, and A.Wolisz, "EvalVid - A framework for video transmission and quality evaluation", Int'l Conf. on Modeling Techniques and Tools for Computer Performance, pp. 255-272, 2003.