Adaptive Rateless Coding for IPTV 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. 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.**

Keywords—IPTV, rateless channel coding; video streaming; WiMAX

I. INTRODUCTION

The BBC's iPlayer in the UK [1] has demonstrated the demand for 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. 802.16e 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 caused by slow and fast fading on a WiMAX channel.

The prior IPTV Content Distribution Network 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. It may also be possible to include limited retransmission of extra redundant data, made feasible through rateless coding.

The iPlayer, as mentioned above, is a simple simulcast service with H.264/AVC (Advanced Video Coding) codec rates available at 500 kbps, 800 kbps, and 1500 kbps, which once selected are fixed to the capacity of the access network. 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 [8], 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 [9] for 3GPP's Release 6 Multimedia Broadcast/Multicast Service (MBMS). Unfortunately, there is no feedback control channel [10] 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 [11] is employed as a means of graceful degradation according to channel conditions. Another approach [12] 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 [12] is not adaptive either. The work in [13] explored the possibility of multiple sources generating Raptor code independently of each other to protect layers within scalable video coding. The paper investigated the cordination of the scheme to achieve rate-distortion optimization. In fact, perhaps the nearest scheme to ours provided for Internet video streaming [14]. 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 [15] 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. If UDP-lite [15] rather than UDP transport is used for streaming then corrupted packets can be passed up the protocol stack to the application layer, as UDP-lite provides partial checksums covering only part of the data such as the header.

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 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. The following Section now describes the adaptive rateless scheme.

II. 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 Gilbert-Elliott model [16], which is a form of hidden Markov model with a good and bad state. If *PGB* and *PBG* are the probabilities of going from the good to bad state and from going from the bad to good state respectively, then

$$\pi_G = PBG/(PBG + PGB) \tag{1}$$

$$\pi_B = PGB/(PBG + PGB) \tag{2}$$

are the steady state probabilities of being in the good and bad states. Consequently, the mean probability of loss is given by

$$PL_{mean} = PG.\ \pi_G + PB.\ \pi_B \tag{3}$$

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.

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

$$R = L \times PL \times A \tag{4}$$

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:

$$\begin{aligned} R &= L\times PL+(L\times PL^2)+(L\times PL^3)\ldots \\ &= L/(1-PL) \end{aligned} \tag{5}$$

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 modeled 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 [17] was employed:

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

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 [17] 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 (7) 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.

The following Section describes the experimental methodology employed to derive the results evaluated in Section IV. Section V concludes this paper.

III. METHODOLOGY

A. Detecting errors

A corrupt packet can be detected by the Cyclic Redundancy Check (CRC) that is an optional part of a MAC Protocol Data Unit (MPDU) (WiMAX packet), refer to Fig. 1. 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 (7) that if less than *k* symbols (bytes) in the payload might be successfully received then $k\text{-}m'+e$, $e > 0$ redundant bytes can be sent to reduce the risk of failure. In tests, $e = 4$, resulting in a 9% risk of failure because of the exponential decay of the risk evident from equation (6). The extra data are additional data over and above the adaptively-estimated redundant data originally added to the packet.

To reduce 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. 2 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.

B. *WiMAX simulation configuration*

To establish the behavior of rateless coding under WiMAX the well-known ns-2 simulator augmented with a module [18] that has proved an effective way of modeling IEEE 802.16e's behavior.

For the Gilbert-Elliott model parameters, *PGG* (probability of remaining in a good state) was set to 0.95, *PBB (*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) = 0.165. Effectively, this model emulates fast fading between good and bad conditions.

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 Time Division Duplex (TDD) frame length was set to 5 ms, as this is the value specified by the WiMAX forum. 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 subscriber station (SS). 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 SS. 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. Sender buffer sizes were set to fifty packets (WiMAX MAC Protocol Data Units). We are aware that this congestion configuration is one of many and future work should investigate this dimension of the problem.

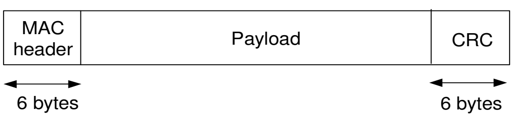


Figure 1. General format of a MAC PDU with optional CRC.

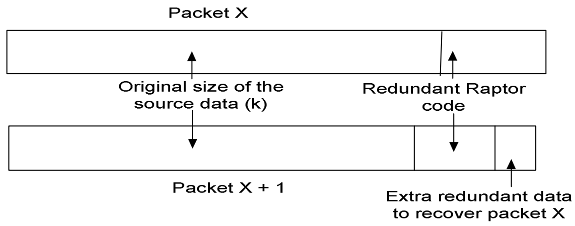


Figure 2. Division of payload data in a packet (MPDU) between source data, original redundant data and piggybacked data for a previous packet.

TABLE I. 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 SS	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, SS = subscriber station

C. *Video configuration*

The JM 14.2 version of the H.264/Advanced Video Coding (AVC) codec software was employed with the Evalvid environment [19] 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*'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 spikes associated with period I-frame. 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.

As a simple form of what is, in effect, layered coding, data-partitioning [11] was enabled within H.264/AVC. In data-partitioning each slice within a picture in the compressed

[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.

domain is split into partition's A, B, and C. Each partition is output as a Real Time Protocol (RTP) packet, encapsulating an H.264/AVC Network Adaptation Layer Unit (NALU). By partitioning in this way, important parameter settings and motion vectors (MVs) are contained in partition-A packets. Partition-B packets contain intra-coded data, both naturally coded intra MBs and the intra-refresh MBs mentioned earlier. Finally, partition-C packets contain texture information (transform coefficient residuals). The latter were compensated for (if lost) by motion-copy error concealment at the decoder using the MVs in partition-A. Thus, data-partitioning can act as a coarse-grained for a graceful degradation. For high- to medium-quality[2] video the size of partition-A is smaller in length than the other two partitions and, therefore, less likely to suffer from channel error. Similarly, partition-B is generally smaller than partition-C, provided that the intra-refresh MB contribution is kept to a low percentage (as herein).

IV. EVALUATION

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 II shows some sample values of factor *A* from equation (4) found during these investigations.

There are two potential gains from applying the adaptive scheme from formula (5), rather than a fixed factor *A,* according to Table II. The first is that Table II's values can only be arrived at through a considerable number of tests for a particular scenario. The second gain is that Table II'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 redundnat 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. 3, the percentage of corrupted packets is

TABLE II. 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

[2] In H.264/AVC, variable bit-rate video quality can be regulated by setting the quantization parameter (QP), with the H.264/AVC QP range being 0–51 (high to very low quality).

recorded according to a variation in the value of QP and increasing probability of data loss in the bad state, *PB*. Recall from Section III.C that varying QP changes the ratio between the data-partitioning packet sizes as well as changing the overall size of the compressed data for any one picture.

The Raptor code equation (6) was applied to decide if a packet could be recovered, given the number of bytes that were declared to be in error. In these tests, the buffer size was sufficient to avoid any packet drops from congestion. Also to simplify the analysis, no packets were lost completely in the channel, because the signal was always assumed to be strong enough for packet reception. These conditions are unlikely to be met in practice and, thus, Fig. 3's results represent an upper bound to performance.

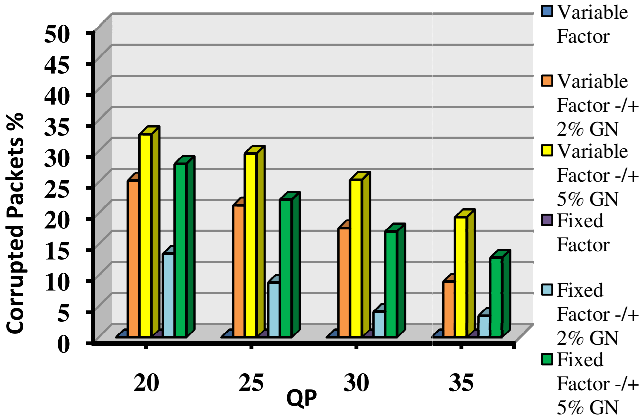
In Fig. 3a, when *PB* = 0.05, assuming no measurement noise results in zero packet loss whether a fixed or variable factor is used in the estimation of additional data. (Zero loss is represented by a flat bar in Fig. 3a.) This is because the exact fixed factor for this *PB* has been selected. For the 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. 3b and 3c. Thus, in Fig. 3b and 3c, 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. 3b and 3c represents the situation when a mis-estimate of this factor has occurred. From Table II it can be seen that the mis-estimate may only need to be by a small percentage before weaker performance results than the variable factor scheme.

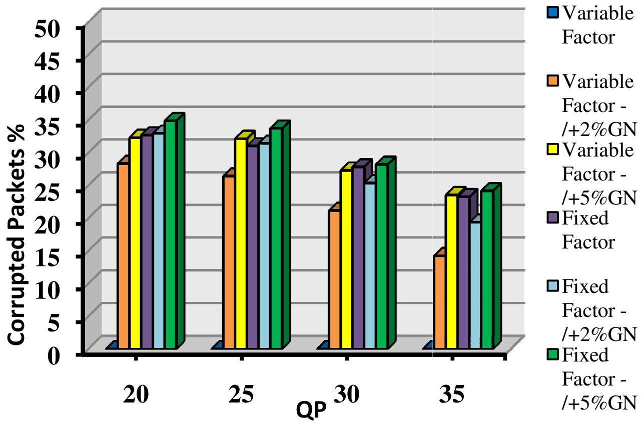
Table III shows the resulting objective video quality and the packet delay, for the packets that did not receive retransmitted data before they were passed to the decoder and those that did (corrupted packets). Though Table III is for *PB* = 0.05, variable factor adaptive scheme with 2% additive Gaussian 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.

V. CONCLUSION

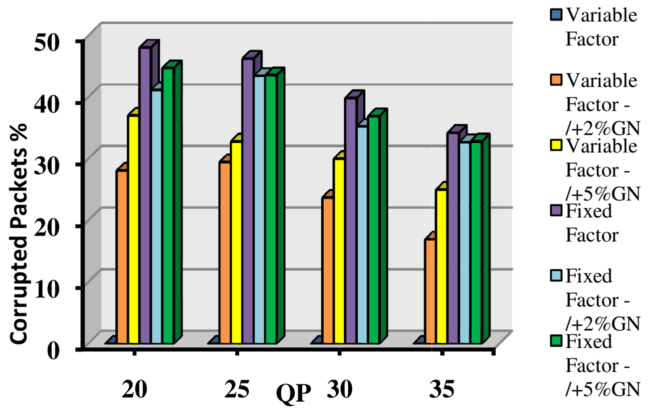
Two adaptive rateless channel coding schemes were presented. To reduce the number of corrupted packets it is possible to empirically estimate the redundant data quantity that will minimize the corrupted packet number (assuming some measurement noise in estimating the packet loss rate).



(a) *PB* = 0.05



(b) *PB* = 0.10



(c) *PB* = 0.15

Figure 3. Variable versus fixed factor adaptive rateless coding.

TABLE III. 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

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. Finally, whichever scheme reduces the number of corrupted packets reduces the overall delay introduced into the video stream.

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

[9] 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.

[10] C. Hellge, T. Schierle, J. Husch, T. Rusert, M. Kampman, and T. Wiegand, "Graceful degradtion 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.

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

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

[13] 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," in *Proc. IEEE Int'l Conf. on Image Processing*, pp. 497–500, 2007.

[14] 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.

[15] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "The lightweight user datagram protocol (UDP-Lite)," IETF, RFC 3828, Jul. 2004.

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

[17] 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.

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

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

[20] 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.