Computers **2015**, *4*, 113-141; doi:10.3390/computers4020113

OPEN ACCESS

computers

ISSN 2073-431X

www.mdpi.com/journal/computers

Article

Error and Congestion Resilient Video Streaming over Broadband Wireless

Laith Al-Jobouri [1], Ismail A. Ali [2], Martin Fleury [1,]* and Mohammed Ghanbari [1]

[1] School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; E-Mails: lamoha@essex.ac.uk (L.A.-J.); ghan@essex.ac.uk (M.G.)

[2] Department of Electrical and Computer Engineering, University of Duhok, 1006 AJ Duhok, Iraq; E-Mail: iaali@uod.ac

* Author to whom correspondence should be addressed; E-Mail: fleum@essex.ac.uk;
Tel.: +44-1206-872-770; Fax: +44-1206-872-900.

Academic Editor: Aaron Quigley

Received: 13 January 2015 / Accepted: 14 April 2015 / Published: 21 April 2015

Abstract: In this paper, error resilience is achieved by adaptive, application-layer rateless channel coding, which is used to protect H.264/Advanced Video Coding (AVC) codec data-partitioned videos. A packetization strategy is an effective tool to control error rates and, in the paper, source-coded data partitioning serves to allocate smaller packets to more important compressed video data. The scheme for doing this is applied to real-time streaming across a broadband wireless link. The advantages of rateless code rate adaptivity are then demonstrated in the paper. Because the data partitions of a video slice are each assigned to different network packets, in congestion-prone wireless networks the increased number of packets per slice and their size disparity may increase the packet loss rate from buffer overflows. As a form of congestion resilience, this paper recommends packet-size dependent scheduling as a relatively simple way of alleviating the buffer-overflow problem arising from data-partitioned packets. The paper also contributes an analysis of data partitioning and packet sizes as a prelude to considering scheduling regimes. The combination of adaptive channel coding and prioritized packetization for error resilience with packet-size dependent packet scheduling results in a robust streaming scheme specialized for broadband wireless and real-time streaming applications such as video conferencing, video telephony, and telemedicine.

Keywords: broadband wireless; access network congestion; data partitioning; multimedia networking; packet scheduling

1. Introduction

Despite increases in capacity, broadband wireless systems still suffer from limitations in bandwidth capacity [1], leading to congestion at a base-station's (BS's) output buffer to its mobile stations (MSs). To reduce the risk of buffer overflow, this paper proposes a customized packet-scheduling regime for data-partitioned video streams. However, taking measures to reduce the impact of congestion at a BS buffer is only one part of the story of how to achieve resilient video streaming over wireless links, which are prone to error bursts [2], leading to a lack of synchronization between video encoders and decoders. In fact, the proposed packet-scheduling regime takes place within the context of a video streaming system that makes use of two measures to counteract the impact of errors: (1) source-coded error resilience [3], *i.e.*, within the video codec, in the form of data partitioning [4]; and (2) channel-coded adaptive Forward Error Correction (FEC) combined with error-control signaling [5]. However, these measures are powerless against packet drops through buffer overflow. Unfortunately, due to the predictive nature of video coding [6], most packet losses also have an effect that extends in time until the decoder is reset (intra refreshed).

One source-coded error resilience scheme, data partitioning, in its current form, is vulnerable to congestion due to the number of small packets that are produced. Data partitioning, which is a form of layered error resilience, can provide graceful degradation of video quality and, as such, has found an application in mobile video streaming [4]. In an H.264/Advanced Video Coding (AVC) codec [7], when data partitioning is enabled, every slice is divided into three separate partitions: partition-A has the most important data, including motion vectors (MVs); partition-B contains intra coefficients; and partition-C contains inter coefficients, the least important data in terms of reconstructing a video frame at the decoder. These data are packed into three types of Network Abstraction Layer units (NALUs) output by the codec. The importance of each NALU-bearing packet is identified in the NALU header. Though it is possible to aggregate (or segment) NALUs [8] before encapsulation in Internet Protocol (IP)/User Datagram Protocol (UDP)/Real-time Transport Protocol (RTP) packets, when data partitioning is in use, this would be to neglect the advantages of retaining smaller partition-A and -B bearing packets because smaller packets have a lower probability of channel error. Therefore, in this paper each NALU is assigned to its own packet, prior to the addition of network protocol headers.

To reduce the risk of buffer overflow for data-partitioned video streams, this paper proposes a packet-scheduling method, which can work in addition to other channel error protection methods, one of which is the adaptive rateless channel coding that is described in Section 2. Packet-scheduling schemes, which are referred to in Section 3, act irrespective of physical-layer (PHY-layer) data scheduling and may be independent of the data-link Medium Access Control (MAC) sub-layer (though some other schemes do indeed intervene at the MAC sub-layer). The proposed packet-scheduling method works by smoothing the packet-scheduling times across one or more video frame intervals, according to allowable latency. Packets are allocated a scheduling time for output to a BS buffer in proportion to their size.

As such, the scheduling method is relatively simple to implement, which is one of its attractions owing to the need to reduce latency for interactive services such as mobile video-conferencing and video telephony. In the paper, the potential value of the application-layer (AL) scheduling approach is demonstrated in simulations that take account of PHY-layer packetization and scheduling. The paper also comments on video-content-dependent packetization issues as an aid to others planning AL packet scheduling or video-smoothing algorithms [9]. Alongside the scheduling regime, the paper also presents the gains from employing adaptive FEC as opposed to statically determining the redundancy overhead. In fact, adaptive rateless FEC is a significant feature of the proposed scheme. For codec-dependent aspects of the scheme, the reader is referred to [10] by the authors.

In general, the contribution of this paper is that a resilient video streaming scheme needs to consider both error resilience and congestion resilience, as error resilience alone is unable to protect against packet drops before the wireless channel is even reached. Section 2 describes the design decisions made by the authors in arriving at the video streaming system, such as the decision to employ rateless channel coding, which at the application layer of the protocol stack has largely superseded [11] fixed rate codes, especially those codes with high decoding complexity. Section 3 of the paper critically reviews recent work on packet scheduling, especially for use in multimedia applications. Section 4 details the methodology of both the error resilience and the congestion resilience aspects of the video streaming scheme. This is followed in Section 5 by an evaluation of selected aspects of the proposed scheme, as simulated for a Worldwide Interoperability for Microwave Access (WiMAX) wireless link, especially the performance of adaptive, rateless coding and the video-content response for packet scheduling. Finally, Section 6 draws some conclusions about the proposed video-streaming scheme.

2. Achieving Error Resilience: Design Decisions

Robust unicast communication can be achieved by two means of error control [12,13]: (1) Automatic Repeat ReQuest (ARQ) and (2) FEC. ARQ allows a higher video data throughput than FEC provided there is a feedback channel and provided the channel is not inherently error prone. ARQs are also much simpler to process than FEC. However, as in the example of this paper, ARQ and FEC are not mutually exclusive: they can be employed together.

In best-effort wired networks, in order to add reliability to an unreliable protocol such as UDP, ARQ at the application layer can be achieved by Acknowledgments (ACKs), assuming the ACK latency does not impact upon the video display rate [14]. For example, in [15] the transmitter keeps a buffer of packets containing video frames, some of which have been transmitted but not acknowledged and others of which have yet to be transmitted. The transmitter must ensure that the buffer occupancy level does not exceed the potential transmission delay. Cyclic Redundancy Checks (CRCs) are required (and assumed in this paper) to detect errors, if the data has not already been so checked at the transport layer. For example, the UDP-Lite protocol [16] allows, when desired, data with errors to be passed up the protocol stack. UDP Lite is a possible implementation route for the ARQ/FEC system in this paper, allowing data to be corrected at the receiver rather than lost to the wireless receiver.

In a broadband wireless link designed for multimedia traffic such as the WiMAX link of this paper, a communication frame [17] is divided into two sub-frames: the first from the base station to mobile stations and the second from mobile stations to use for communication with the base station. Therefore, a natural way of delivering an AL ACK is by placing it in the return sub-frame, as we do in this paper. Notice that WiMAX also permits data-link layer ACKs and, alternatively, even a form of hybrid ACKs to be turned on on a per-link basis. However, switching ACKs on at the MAC sub-layer may result in arbitrary delays that are beyond the control of a multimedia application, which is why they were not enabled in this paper's evaluations. Instead AL ACKs were sparingly utilized, namely just once per packet. Soft ACKs can also counter the problem of arbitrary delays resulting from data-link ACKs by limiting the delay to a maximum [18], but there is a risk of poor-quality video. To reduce that problem, the source coding rate [19] can be varied according to the potential delay. However, in designing our system, it was simpler to implement AL ACKs.

To avoid the problem of delay and possible resource consumption by ACKs, FEC-based approaches transmit source-coded data with additional redundant parity data. Traditional fixed-rate channel codes cannot dynamically adapt to changing channel conditions easily, while rateless codes, with changing ratios of redundant to information data, can now adapt in a graduated fashion. However, to enable adaptation, Hybrid-ARQ (H-ARQ), e.g., as in [20], becomes necessary. In Type II H-ARQ, the transmitter only retransmits the necessary redundant data to increase the probability that the received data can now be reconstructed. In this paper, Type II H-ARQ is used. Moreover, as previously mentioned, to limit the impact of ARQs on latency, only one request for additional FEC is permitted.

Adaptive FEC with hybrid H-ARQ is still possible with fixed rate codes, but some other means of rate adaptation is required such as code puncturing [21] (the removal of parity bits) or code extension (the addition of parity bits resulting in a slower code rate). Unfortunately, some codes, for example Reed-Solomon (RS), are known to consume battery power [22], caused ultimately by their asymptotic (decoding) complexity which is O(k^3) for the Gaussian elimination algorithm and O(k^2) for the Berlekamp-Massey algorithm, where k is the number of information symbols. Therefore, due to their computational complexity, fixed-rate codes were avoided by us. Instead, Raptor codes [23], the variety of rateless code employed herein, have linear complexity both for encoding and decoding. They permit decoding if any k encoded symbols successfully arrive at the decoder. Usually, a small percentage (approximately 5%–10%) of additional encoded symbols are transmitted for successful recovery, but in the case of the requirement for additional encoded symbols, these additional symbols can be generated by the rateless encoder, in what has been called the fountain approach [24].

Raptor codes are one of a number of concatenated codes, e.g., the Turbo-Fountain code [25], which historically were developed to reinforce Luby Transform (LT) codes [26] when it was pointed out that LT codes had high error floors [27]. High error floors imply that the risk of decoder failure does not fall away as the channel conditions improve; instead, the risk remains at the same level as for lower Signal-to-Noise Ratios (SNRs). In order to achieve this, linear-time decoding iterative belief-propagation algorithms are necessary, which can be applied both to Raptor's outer code, a variant of Low-Density Parity Check (LDPC), and to the LT inner code. Furthermore, a systematic Raptor code can be achieved by initially applying the inverse of the inner code to the first k symbols prior to the outer coding step, which operates on the first k information symbols and an additional set of redundant symbols. Systematic channel codes separate the redundant coding data from the information data, allowing the information data to be passed directly to the decoder if no errors are detected (usually by means of CRCs). However, finding the inverse via Gaussian elimination increases the time complexity. Notice that in practice [28], Raptor codes, despite their low theoretical time complexity, are only computationally effective if an appropriate outer code is chosen, originally LDPC and latterly two concatenated LDPC-like codes.

In a traditional packet erasure rateless coding scheme, if PHY-layer correction fails due to checksum detection [29], then a packet becomes an erasure to be corrected by the rateless code at the application layer. However in our paper, upon PHY-layer correction failure, packets are not marked as erased but their data are passed to the application layer for Raptor code correction. This is possible because, in our paper, the information symbol is no longer an erased packet but a data block within a packet (intra-packet). (Notice that in the Multimedia Broadcast Multicast Service (MBMS) implementation of rateless codes [30], blocks are in general inter packet, not intra packet. However, this arrangement has the potential to increase the organizational overhead and the latency of the decoding process.) For a real-time multimedia application, intra-packet symbols have the additional advantage that there is no longer a requirement to wait for k packets to successfully arrive; k data blocks within a packet can arrive. In a procedure introduced in [31], each data block is assigned a checksum. If the checksum calculation fails, that block is then marked as an erasure. As an alternative, also in [31], each block can be given a confidence value based on the log-likelihood ratio (LLR) of the bits within it. Thus, the reduced risk of decoder failure is traded off both against lower latency and also the reduced efficiency of the code. The latter is due to the risk of declaring as erased blocks that are actually valid. In our intra-packet scheme, the data block is reduced to the size of a single byte, which has the advantage that the risk of decoding failure (see next paragraph) from shortage of data symbols within a packet is reduced. However, this raises the issue of CRCs, as the overhead would be too much if each byte were to be protected by a CRC. Therefore, a hybrid scheme is more suitable, in which blocks of bytes are assumed to be protected by a single CRC and marked as erasures if the CRC fails.

In a Raptor code for the inner LT code, even in an error-free channel, there is a small probability that the decoding will fail. However, that probability converges to zero in polynomial time in the number of input symbols [32] and a similar analysis applies to an outer LDPC [33]. In this paper, the probability of decoder failure is modeled statistically by the following equation from [34]:

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

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 for k > 200 [34] the model of Equation (1) almost ideally models the performance of the code. This implies that, if block symbols are used, approximately 200 blocks should be received before reasonable behavior takes place. Therefore, we require packets of at least 200 bytes in our scheme.

In summary, the resulting video communication system used by us over a WiMAX link has protection at several layers of the protocol stack. Data randomization is applied at the PHY layer to avoid runs of 1 s or Os. Subsequently, one of WiMAX's modulation and coding schemes is selected. Mobile WiMAX offers modulation by one of Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (QAM), or 64-QAM in descending order of robustness. The mandatory convolutional coding rate is selected from 1/2, 2/3, 3/4, and 5/6 in ascending order of protection.

There are other WiMAX PHY-layer protection options such as Turbo coding or PHY-layer H-ARQ which were not configured in the simulation experiments of Section 5. Thus, bit-level FEC protection is first applied at the PHY layer and additional FEC is applied in our scheme at the application layer by means of a rateless channel coder. However, unlike conventional block-based AL channel coding, a variant of packet-level channel coding, we used intra-packet byte-level rateless channel coding. By selecting this form of application-layer channel coding, we were able to use adaptive rateless coding enabled by means of a simple, low-latency H-ARQ mechanism.

3. Related Work

3.1. Rateless Code for Video Streaming

Prior usage of AL Raptor coding, especially that in various wireless standards [35], such as DVB-H and 3rd Generation Partnership Project (3GPP)’s Multimedia Broadcast Multicast Service (MBMS), has been multicast [36] without feedback, rather than unicast with feedback as herein. Moreover, unlike herein, it has been deployed in the standards with block-level symbols that can extend across packets rather than intra-packet as herein, resulting in longer repair latencies if and when the blocks extend over many packets.

Data-partitioned video streams were first protected by a form of Raptor codes in [37], which paper from the current author also demonstrated that data partitioning has the least overhead of H.264/AVC error resilience methods. Since then, the idea was taken up in [38] providing an application to DVB-H and comparing with random linear codes (RLC). It was demonstrated that this form of AL-FEC is effective at lower data rates when compared to H.264/AVC without data partitioning. RLC were competitive with rateless coding in terms of error protection. However, as the information data length increases, the need to employ a Gaussian elimination (rather than belief propagation) decoding algorithm causes the computational complexity and, hence decoding latency, to increase as O(n^3) rather than O(n) for Raptor codes. In [39], RLC were applied in the same form as [37] had already used for Raptor codes, that is with expanded windows code (called Growth codes in [37]) to provide scalable channel coding. However, in [39], the protection was not applied to data-partitioned video but by prioritizing the different H.264/AVC slice types. Finally, there is a summary of the previous investigations of [38,39] in [40]. The current contribution does not pursue the ideas originally developed by us in [37] because we seek herein to show that the complication of expanding window or growth codes may be unnecessary for data-partitioned video streaming.

In our paper, adaptive provision of redundant rateless code results in reduced channel coding overhead. It is possible that if further advantage of cross-layer communication could be made then the overhead of rateless coding can be reduced still further. For example in [41,42], the amount of rateless data was jointly determined by the modulation and coding scheme (MCS) at the PHY layer. It is possible that adaptation according to the wireless channel state could be combined with adaptation according to the PHY-layer MCS mode in a future cross-layer scheme.

3.2. Packet Scheduling for Video

In [43], the authors combined scheduling at the transmitter with control of playout speed at the receiver, across a time-varying wireless link. This proposal required video-content-dependent decisions with the aim of maximizing video quality at the receiver. Decisions were driven by a Markov process informed by the motion activity of the transmitted video frames. In the scenario of [43], the scheduler discards late packets and catches up with delayed playout during poor wireless channel conditions. The receiver adapts its playout rate (the rate video frames are displayed at), slowing it down when playing video with low motion and, hence, low coding complexity occurs. The aim was to avoid playout interruptions, which are thought to be less disconcerting to the viewer than small slowdowns in the playout rate. The authors concluded their research by identifying the need for practical heuristics that reduce the complexity of their proposed schemes due to the need to inspect the content type, which remains a problem for real-time operation and is the specific problem our simplified scheme seeks to address.

The optimal scheduling decision is again computed through a Markov process in [44]. However, the authors of [44] are aware that burst errors are a threat and, in particular, may disrupt feedback signals. Therefore, modeling is designed to take into account incomplete information about the receiver’s state. The end result is improved video quality but delays to video packet transmission during poor channel states. In this early work, the impact on playout delay was not taken into account and the authors also acknowledge the need to develop heuristic decision-guiding rules.

In [45], the video content is taken into account to vary the packet scheduling decision at the transmitter. Inspection of the video content does imply an increase in end-to-end delay, as does the need to await protocol acknowledgments. The position of a packet’s data within a Group-of-Pictures (GOP) weights its importance to the playout process and, hence, determines its transmission schedule deadline. Similarly, the texture- and motion-complexity governed scheduling decisions when adapting the packet transmission order. Simplified wireless conditions were assumed to test the scheme. However, the authors note that channel coding is insufficient in itself during “bursty” error conditions, requiring intelligent ARQs and/or packet scheduling. The result of packet scheduling was improved objective video and a more even distribution of temporal distortion. Data partitioning was also turned on in some of the evaluation tests but with the Moving Picture Experts Group (MPEG)-4 part 2 form of data partitioning, in which each video packet is internally divided between shape/motion data and less important texture data.

The work in [46] is aimed at IEEE 802.11 Wireless Local Area Networks (WLANs) running the Distributed Coordination Function (DCF) at the MAC layer, *i.e.*, retransmitting data packets that are lost. By determining the importance of the video data within a packet, differing re-try limits occur. The codec includes the importance of the video data as side information with the video data. However, to achieve significant gains in video quality requires cross-layer intervention in the wireless protocol stack as well as modification of the video codec. Research in [47] introduced another scheduling scheme that relies on video content side information placed in advance within the bitstream by the encoder. It is designed in a practical fashion for a multi-user, Time Division Multiple Access (TDMA) wireless environment such as High Speed Downlink Packet Access (HSPDA).

Long Term Evolution (LTE) leaves the choice of packet scheduler to the vendor, resulting in many research proposals [48]. These are system-level schedulers, of which [49] combines the popular proportional fair algorithm with a higher-level scheduler that allocates the amount of data that each real-time flow can send within one time epoch.

4. Methodology

The error and congestion resilient video streaming scheme is demonstrated for IEEE 802.16, which is the standardized version of WiMAX wireless broadband technology [50]. WiMAX continues to be rolled out in parts of the world that do not benefit from existing wired infrastructures or cellular networks. WiMAX is also cost effective in rural and suburban areas in some developed countries. It is designed to provide effective transmission at a cell’s edge by the allocation to a mobile user of sub-channels with separated frequencies to reduce co-channel interference. The transition to the higher data rates of IEEE 802.16 m [51] indicates the technological route by which WiMAX will respond to the technological advances of its competitors, especially LTE. However, we modeled version IEEE 802.16e-2005 (mobile WiMAX), of which IEEE 802.16-2009 is an improved version [52], as this is backwards compatible with fixed WiMAX (IEEE Project P802.16d), which still remains the most widely deployed version of WiMAX. Time Division Duplex (TDD) (Figure 1) and effective scheduling of time slots between MSs through TDMA (not to be confused with AL packet scheduling as described in this paper) increases spectral efficiency. Figure 2 provides a closer look at the frame and sub-frame structure within which are contained per-MS data bursts. (For simplicity, the MS to BS sub-frame structure is illustrated.) Within each data burst are WiMAX packets, which at the MAC sub-layer are called Protocol Data Units (PDUs) and consist of a MAC header, MAC Service Data Unit (MSDU), and a CRC, as shown in Figure 3. An MSDU corresponds to a packet passed to the MAC layer from higher layers in the protocol stack, most immediately from the transport layer. It consists of an IP packet with typical IP/UDP/RTP headers attached, though, in practice, packet header compression is normally employed over wireless links in order to reduce the heard overhead. The payload of this network packet will contain an NALU and additional rateless channel coding data added at the application layer, as described below.

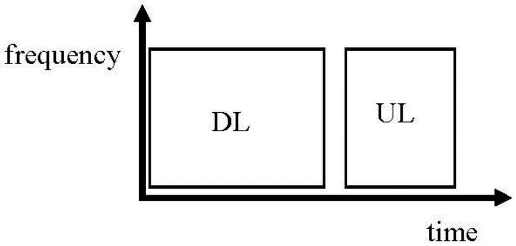


Figure 1. Worldwide Interoperability for Microwave Access (WiMAX) frame divided into two sub-frames separated by a guard interval, namely the downlink (DL) sub-frame from a base-station (BS) to mobile stations (MSs), and an uplink (UL) sub-frame following in time from the MSs to the BS. The time direction consists of successive Orthogonal Frequency-Division Multiple Access (OFDMA) symbols, while the frequency direction is assigned to data bursts on a per-MS basis (OFDMA is a refinement of Orthogonal Frequency Division Multiplexing (OFDM) that in mobile WiMAX allows more flexible sub-channelization in the frequency dimension). (Additional preamble, burst mapping, and ranging signaling is also present. For more details refer to [50].)

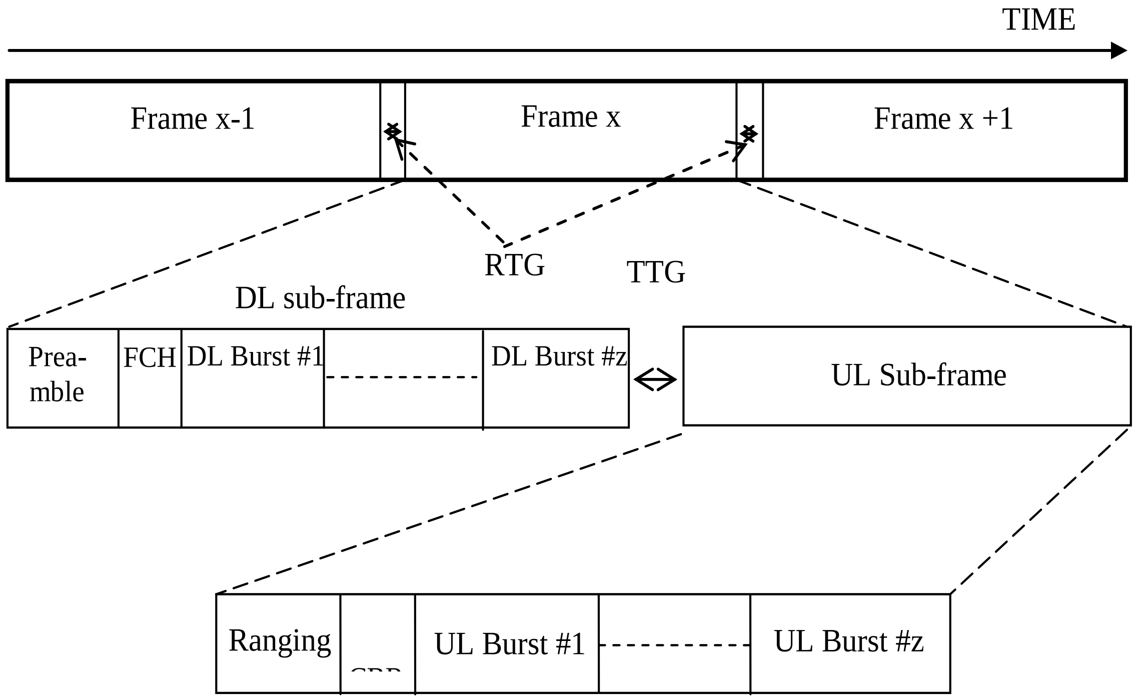


Figure 2. Sub-frame structure showing per-MS data bursts (numbered from 1 to z). (FCH = frame control header, RTG = receive/transmit transition gap, TTG = transmit/receive transition gap.)

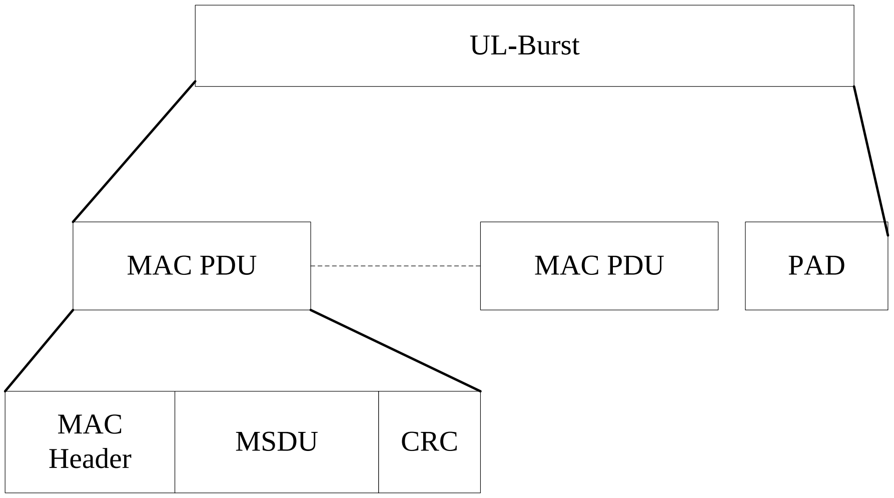


Figure 3. WiMAX burst structure showing MAC Service Data Unit (MSDU) packets. The DL-Burst is similar, except that there are additional headers included prior to the MAC PDUs. (PAD = padding, as the PDU size is variable.)

As a whole, the WiMAX video-streaming scheme, Figure 4, comprises three components: (1) source-coded data partitioning of the video bitstream; (2) adaptive rateless channel coding; and (3) AL packet scheduling. This Section now describes these three components. As analyzed in Section 1, the first two of these are error resilience measures, while the third, supports congestion resilience. Therefore, the following description of the methodology is split into error resilience and congestion resilience. The error-resilience description includes the way that the source packets (NALUs) are mapped to the underlying broadband wireless MAC and PHY layers.

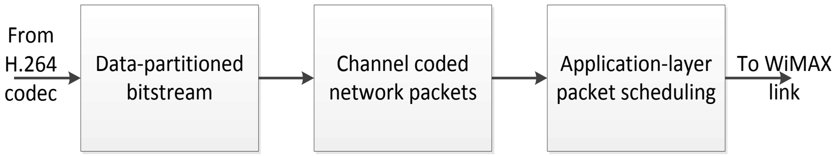

From H.264 codec
Data-partitioned bitstream
Channel coded network packets
Application-layer packet scheduling
To WiMAX link

Figure 4. Overview of video-streaming scheme.

4.1. Error Resilience

As previously remarked, data-partitioned H.264/AVC compressed video can be an effective means of placing more important data (as far as video decoding is concerned) in smaller, less error-prone packets in a wireless channel, provided low-quality video (quantization parameter (QP) greater than 30) is not transmitted, as then the less important packets diminish in size. As is common in mobile video communication, in this paper an IPPPP… frame coding structure is employed, that is an intra-coded (I)-frame followed thereafter by predictively-coded (P)-frames. Notice that bi-predictively-coded (B)-slices are not permitted in the H.264/AVC Baseline profile, aimed at reducing the complexity of bi-predictive coding on mobile devices. Random Intra Macroblock Refresh (RIMR) (forcibly embedding a given percentage of randomly placed intra-coded macroblocks (MBs) in a P-frame) was turned on to counteract spatio-temporal error propagation that would otherwise occur in the absence of I-frames. The advantages of omitting periodic intra-coded I-frames for wireless communication are reviewed in [53]. Figure 5 illustrates the standard frame structure of a codec (called a Group of Pictures or GOP). In the common wireless transmission frame structure, P-frames take the place of the B-frames implying that all prediction is in the forward direction from one P-frame to the next. Moreover, the final I-frame in a GOP is also replaced by a P-frame, as I-frames are now not the means of re-setting the decoder in the event of error corruption or frame drops. The sequence of P-frames now extends to the end of the video sequence or stream. RIMR is employed by us to gradually re-set the decoding process, which implies that there will always be intra-coded MBs within each frame.

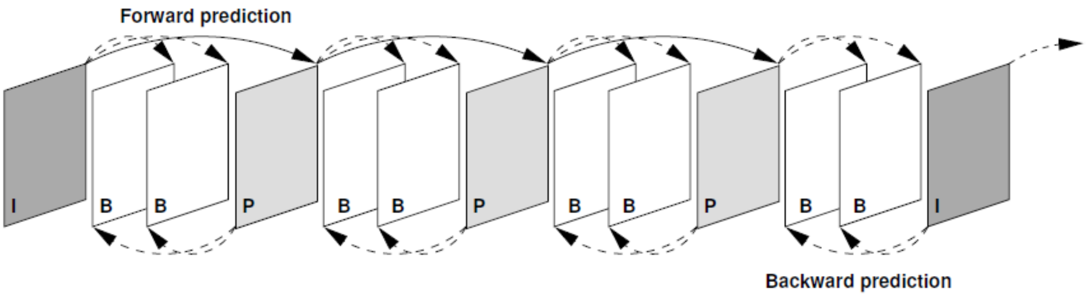


Figure 5. Standard frame structure within a codec.

When data partitioning is enabled, every slice within a video frame (Figure 6) is divided into up to three separate partitions and each partition is encapsulated in either of type 2 to type-4 NALUs (refer to Figure 7). Partition-A, carried in an NALU of type 2, comprises the MB addresses and types, their MVs, and QPs. If any MBs in the frames are intra-coded, their frequency-transform coefficients are packed into a type-3 NALU or partition-B. Partition-C contains the transform coefficients of the motion-compensated inter-coded MBs and is carried in an NALU of type-4. (Partition-B and partition-C also contain Coded Block Patterns (CBPs), compact maps indicating which blocks within each MB contain non-zero coefficients.) Figure 8 illustrates the overall process from the point of view of the H.264/AVC codec processing the raw video data input from a camera. A compressed video stream is output from the Video Coding Layer (VCL) after applying the normal redundancy reduction steps [6] including motion estimation and compensation, transform coding, quantization and entropy coding, which result in a compressed bit-stream. As these steps take place on a per-MB basis, it is possible to separate groups of MBs into slices, which in Figure 6 are formed from geometrically adjacent MBs, selected in raster scan order. The data-partition step separates out the per-MB data by function, e.g., all MVs of the MBs within a slice are separated out. The data-partition step subsequently assigns that data to the appropriate data-partition type. Notice that in Figure 8, one of the slices has no partition-B data, because there are no intra-coded MBs within it. That situation may arise if RIMR MBs do not fall within the slice and also because the encoder does not select any MBs for intra coding, as might happen if an occluded object prevents a prediction match being found. The other two slices have MBs that are spatially-coded (intra-coded). Once partitioning has taken place, the data partitions are allocated to a NALU of the appropriate type at what is termed the NAL of the codec. The FEC redundant data (Raptor code data) are also included (refer to Figure 9) within the packet payload along with checksums. If additional FEC data are requested over the uplink, that data also is included in the packet payload. As NALU aggregation was not enabled, each NALU is assigned to a separate network packet as its payload. Additionally, it is important to notice that, for our experiments, each video frame formed a single slice.

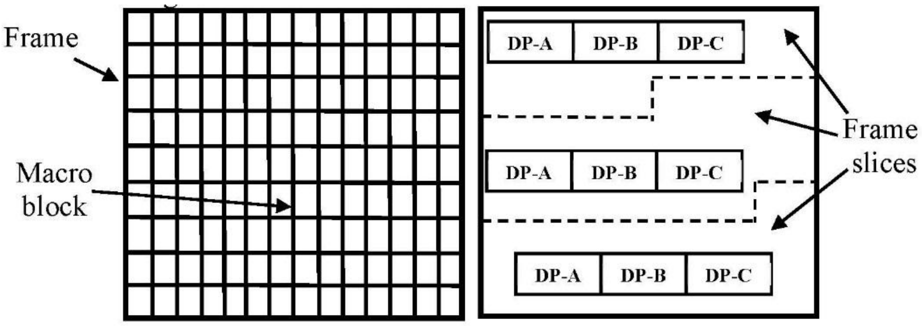


Figure 6. Decomposing a video frame into partitions within slices.

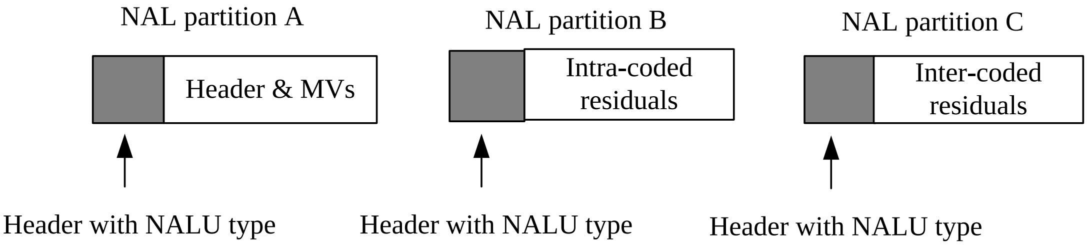


Figure 7. H.264/AVC data partitioning in which the partitions of a single slice are assigned to three NAL units (types 2 to 4). As described in the text, the relative size of the partitions varies with the QP setting of the encoder and is not fixed, as shown in the schematic representation of this figure.

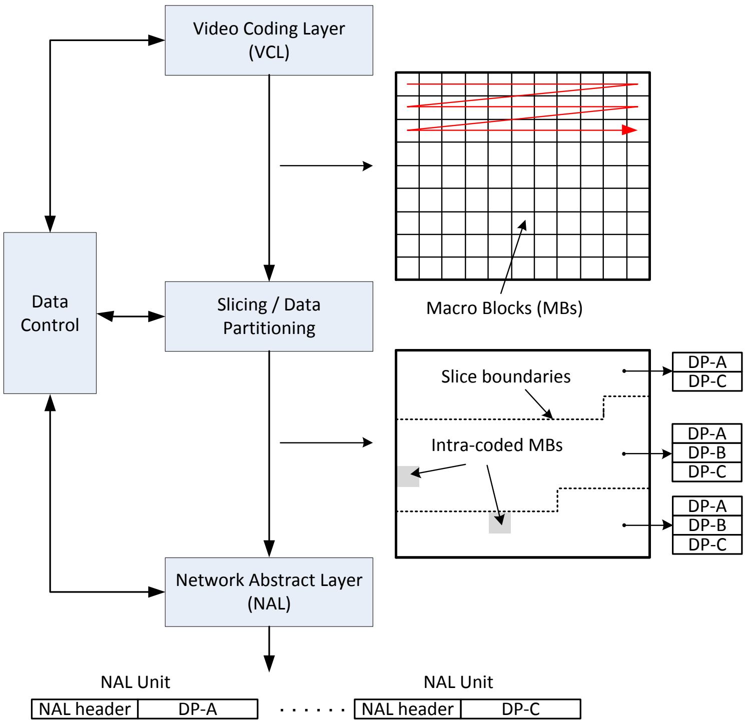


Figure 8. H.264/ Advanced Video Coding (AVC) data flow for combined slicing with data partitioning.

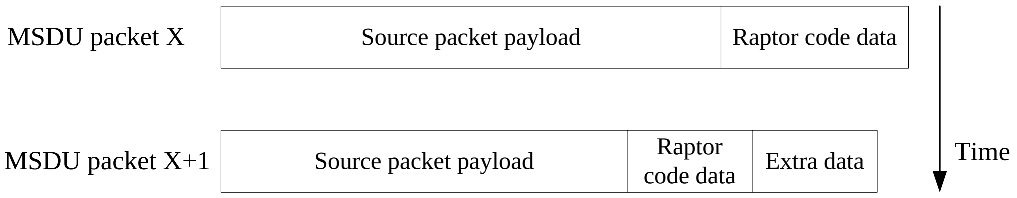


Figure 9. WiMAX packets (MSDUs) from BS to an MS containing video source data from higher layers with additional application-layer-forward error correction (AL-FEC) added by the streaming application. Packet x + 1 contains additional piggybacked AL-FEC data requested to aid in recovery. The Raptor code data overhead is adapted to channel conditions. The amount of code data added is schematic only, as in practice around 5% would be present. Not shown are possible cyclic redundancy checks (CRCs) added by the application and Internet Protocol (IP)/User Datagram Protocol (UDP)/Real-time Transport Protocol (RTP) headers that would also be present in the source packet payload in addition to the Network Abstraction Layer units (NALU).

Reconstruction of the other partitions is dependent on the survival of partition-A, though that partition remains independent of the other partitions. Constrained Intra-Prediction (CIP) [54] (Ref. [54] introduces standardized intra CIP while proposing non-standard inter-CIP) was set in order to make partition-B independent of partition-C. When only partition-A survives, its motion vectors can then be employed in error concealment at the decoder using motion copy. When partition-A and partition-B survive, then error concealment can combine texture information from partition-B when available, as well as intra concealment when possible. When partition-A and partition-C survive, to reconstruct it requires partially discarding partition-C data reliant on missing partition-B MBs and then reconstructing either of them using partition-A MVs or partition-A with partition-C texture data, when it is available.

The relative size of the data-partition packets is determined by the quality of the video, which in turn is governed by the QP of the MBs. The QP is set in the configuration file of the H.264/AVC codec, prior to compression. Figure 10 is a comparison between the relative sizes of the partitions according to QP for the two diverse reference video clips used later in Section 5. For the purposes of assessing the impact of the QP, Variable Bit-Rate (VBR) was encoded in Section 5.1's evaluation. In VBR video, the QP remains constant in order to preserve video quality. However, many broadcasters prefer Constant Bit-Rate (CBR) video, as it allows transmission bandwidth and storage to be predicted. Transmission jitter is also reduced. For that reason, CBR video is tested in Section 5.2, though the QP value may vary a little [55] in order to maintain a constant bit-rate.

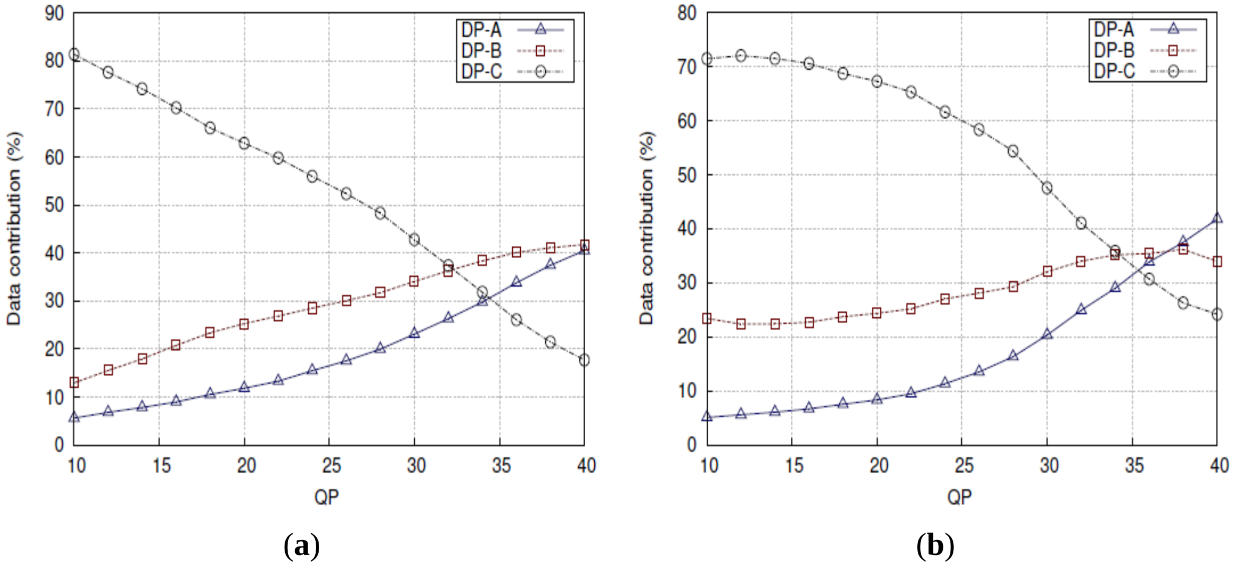


Figure 10. Percentage contributions of data partitions (DPs) A, B and C over a range of QPs for (**a**) *Paris* and (**b**) *Stefan* video clips.

Rateless Raptor code is an effective form of FEC that provides protection to the data partitioned video stream with reduced computational complexity and capacity approaching overhead. Rate adaptation of AL FEC is performed in order to match wireless channel conditions. The code was applied at the level of bytes within a packet in the interests of reduced latency rather than at the usual packet level. Thus, the byte forms the block symbol size and only bytes passing the CRC at the PHY layer are accepted at the application layer channel decoder. The probability of channel byte loss (*BL*) serves to predict the amount of redundant data to be added to the payload. In implementation, *BL* is found through measurement of channel conditions, which is mandatory anyway in a WiMAX mobile station. If the original packet length is *L*, then the redundant data is given simply by

$$R = L \times BL + (L \times BL^2) + (L \times BL^3)\ldots = L/(1 - BL) - L \quad (2)$$

which is arrived at by adding successively smaller additions of redundant data, based on taking the previous amount of redundant data multiplied by *BL*. The statistical model of Equation (1) in Section 2 was utilized to determine the Raptor code failure probability.

Packets with “piggybacked” repair data are also sent. Suppose a packet cannot be decoded despite the provision of redundant data. It is implied from Equation (1) that if less k symbols (bytes) in the payload are successfully received, a further $k - m + e$ redundant bytes can then 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 (1), which gives rise to Raptor code’s low error probability floor. In practice, $e = 4$, reduces the probability of failure to decode to 8.7%. Only one retransmission over a WiMAX link is allowed to avoid further increasing latency. If that retransmission fails to allow reconstruction, that packet is abandoned.

4.2. Congestion Resilience

With respect to congestion resilience, consider Figure 11 in which a single video frame has been assigned equal bit length slices. The geometric space taken up by any slice is dependent on the coding complexity of the content. Shaded MBs in Figure 11 represent intra-coded MBs. Each slice was further partitioned in source coding space into up to three partitions. As mentioned already, it is possible that partition-B may occasionally be absent (the top slice in Figure 11) if no RIMR MBs are allocated to a slice and if no naturally intra-coded MBs are assigned to the slice by the encoder. (Naturally encoded intra MBs are inserted if an encoder can find no matching MB in a reference frame or as a way of improving the quality.) Similarly, it is possible that if, for example, an extremely high bit rate was allocated to the stream, partition C might not be present, as the encoder could afford the luxury of encoding all MBs with intra coding. However, the simple packet scheduling scheme for data-partitioned streams is independent of source-coding allocations of partitions and slices.

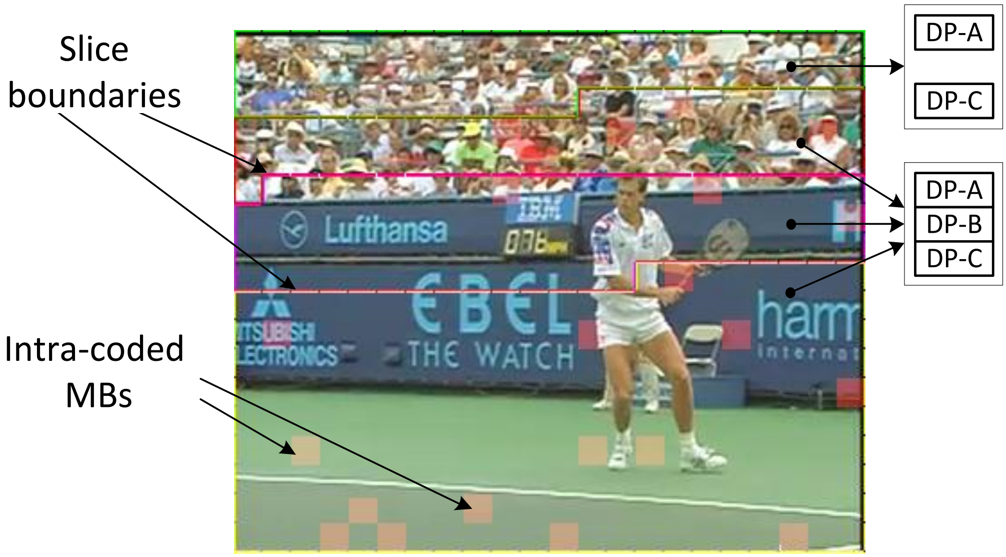


Figure 11. Example video frame from *Stefan* split into equal bit length slices, showing each slice partitioned into up to three data partitions (DPs) (A, B, C).

Figure 12 demonstrates the results of two alternative packet scheduling regimes. A frame interval is shown, which is 1/30 s at 30 frame/s. In the default case, scheduling is at equal intervals in time. We are aware that packet scheduling may never follow this ideal regime in a processor even with a real-time operating system present. However, the regime acts as a point of comparison. In the simple scheduling scheme proposed, packets are allocated a scheduling point according to their relative size within a frame’s (or multiple frames') compressed size. Thus, for any one packet indexed as *j* with length l^j, its scheduling time allocation is:

$$t_{alloc}^{j} = \frac{l^j}{\sum_{i=1}^{n} l^i} f \tag{3}$$

where *f* is the fixed frame interval, and there are *n* packets in that frame. The denominator of Equation (3) sums the lengths of the packets within frame *j* and allocates a time interval relative to its size (numerator of Equation (3)) relative to the total length of the packets within frame *j*.

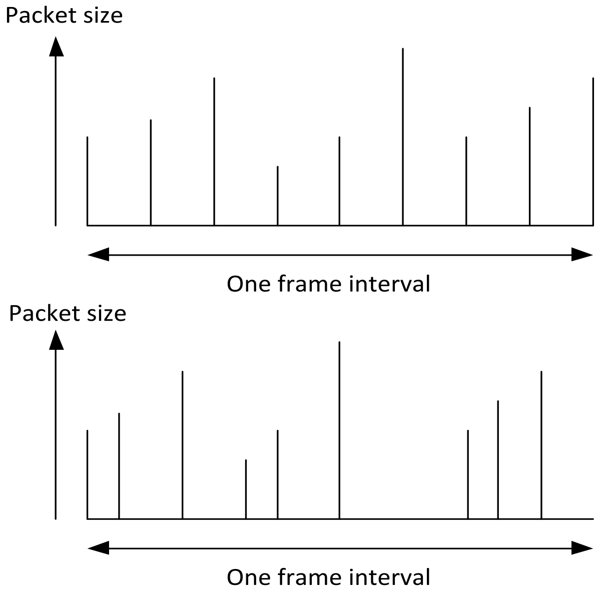


Figure 12. Example of packet scheduling regimes: (**top**) Equally spaced in time; (**bottom**) Packet ordering preserved but scheduling time allocated according to packet size.

The proposed scheduling method obviously preserves the original average bitrate, though there is a single video-frame latency while the frame's packet sizes are assessed and the scheduling takes place. In [56], it is pointed out that IEEE 802.21 Media Independent Handover (MIH) services (IEEE 802.21 WG, 2008) provide a general framework for cross-layer signaling that can be used to achieve the scheduling. In IEEE 802.21, a layer 2.5 is inserted between the level 2 link layer and the level 3 network layer. Upper-layer services, known as MIH users or MIHU communicate through this middleware to the lower layer protocols. For mobile WiMAX and later versions of WiMAX, another WiMAX-specific set of standardized communication primitives exists as IEEE 802.16 g.

Figure 13 identifies the place that AL packet scheduling takes place with the processing cycle and also summarizes the processing cycle as a whole. After compression through the video codec, rateless channel coding is applied at the application layer on an intra-packet basis at the byte level. A simple form of application-layer H-ARQ serves to operate the adaptive form of channel coding. AL packet scheduling now takes place for the source packets. After packetization at the upper protocol stack layers, the PHY layer is responsible for a series of protection measures already mostly described. Data interleaving of each PHY FEC block in an OFDM-based system acts to map adjacent bits across non-adjacent sub-carriers. As previously stated, PHY layer H-ARQ was not enabled in the evaluation of Section 5.

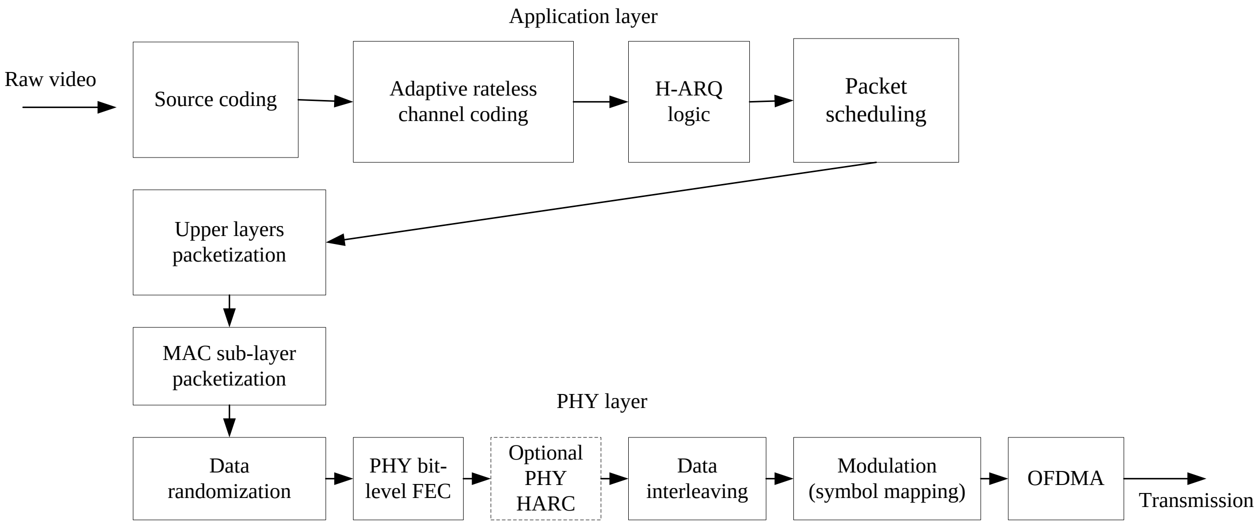


Figure 13. Overview of protection measures and packet handling across the upper and WiMAX protocol stack layers.

5. Evaluation

This section begins by describing the simulation model for the evaluations, while the later two sections concentrate on aspects of error resiliency and congestion resiliency.

5.1. Simulation Model

To model video communication over WiMAX, the well-known ns-2 simulator [57] was combined with a module from the Chang Gung University, Taiwan [58] that has turned out to be an effective way of modeling IEEE 802.16e's behavior. In the evaluation, transmission over WiMAX was carefully modeled. The IEEE 802.16e TDD frame length was set to 5 ms, as previously remarked, because only this value is supported in the WiMAX Forum simplification of the standard (In fact, a frame size of 20 ms benefits video streaming [59] because there is more time to transfer enough data.). The raw downlink data rate of 10.67 Mbps results from the use of one of the mandatory WiMAX modulation and coding modes [50] for a TDD downlink/uplink sub-frame ratio of 3:1, namely 16-QAM at ½ rate. The WiMAX BS was assigned more bandwidth capacity than the uplink to allow the BS to respond to multiple MSs. All buffers were set to 50 packets and all packets were directed to WiMAX's real-time Polling System (rtPS) class of service.

Apart from video streaming to a WiMAX MS, three other MSs received continuous CBR data, Figure 14, with two receiving at 1 kbps, and one at 840 kbps. A typical low bitrate parametric speech codec such as a Linear Prediction Coding (LPC) vocoder operates from 1.2 kbps to 4.8 kbps, close to the lowest rate of these two signals. The G.711 vocoder for high-quality Voice-over-IP (VOIP) has a typical data rate of 87 kbps, implying that the larger of these signals is equivalent to ten such speech signals. Packet size was set at 1 kB, the WiMAX maximum transport unit. It is these streams that inject cross-traffic into the scenario modeled and provide any congestion experienced in the experiments by the video stream.

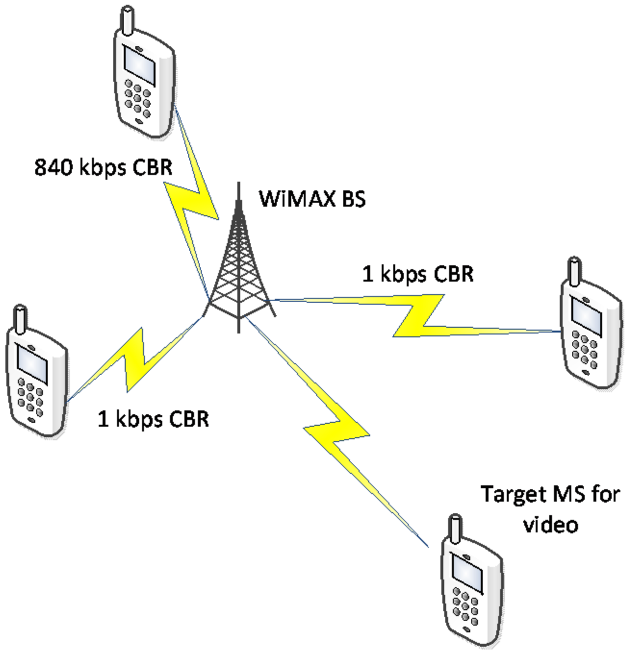


Figure 14. Scenario for evaluation experiments.

To model adverse channel conditions across the WiMAX link, "bursty" errors (time correlated errors) were modeled by a Gilbert-Elliott two-state hidden Markov model [60]. This channel model was introduced into the ns-2 simulator. The Gilbert-Elliott channel model itself is a two-state Markov chain. It is based on good and bad states, the probabilities of these states, and the probabilities of the transition states between them. In the bad state, losses happen with higher probability, whereas in the good state, losses happen with lower probability. P_{GG} refers to the probability of being in the good state and P_{GB} is the probability of a transition from the good state to the bad state. P_{BB} is likewise the probability of being in the bad state and transitioning back to the same bad state. P_{BG} refers to the probability of a transition from the bad to good state. P_{GG} (P_{BB}) can be interpreted as the probability of remaining in the good (bad) state, given that the previous state was good (bad). Conversely, P_{GB} represents the probability that, given that the previous state was good, a transition is made from the good to the bad state. By the law of total probability, all probabilities sum to one (certainty). Therefore, we have $P_{GG} + P_{GB} = 1$, resulting in Equation (4). A similar argument for the bad state leads to Equation (6).

$$P_{GG} = 1 - P_{GB} \tag{4}$$

$$P_{BB} = 1 - P_{BG} \tag{5}$$

For the stochastic process to remain stationary in time,

$$\pi_{\mathrm{G}} P_{GB} = \pi_{\mathrm{B}} P_{BG} \tag{6}$$

where π_{G} and π_{B} are the steady state probabilities of being in a good or bad state respectively. The law of total probability $\pi_{\mathrm{B}} = 1 - \pi_{\mathrm{G}}$ again applies. Substituting this expression for π_{B} into Equation (6) easily leads to:

$$\pi_G = \frac{P_{BG}}{P_{BG} + P_{GB}} \tag{7}$$

Similarly, $\pi_G = 1 - \pi_B$. Substituting this expression for π_G into Equation (6) easily leads to:

$$\pi_B = \frac{P_{GB}}{P_{BG} + P_{GB}} \tag{8}$$

Thus the average loss rate produced by the Gilbert-Elliott channel model is

$$L = p_G.\pi_G + p_B \pi_B \tag{9}$$

where p_G and p_B are the error rates of the good and bad states respectively.

To model the effect of slow fading [61] at the packet-level, as the Gilbert-Elliott model's parameters, the P_{GG} was set to 0.96, P_{BB} = 0.95, p_G = 0.01 and p_B = 0.02. Additionally, it is still possible for a packet not to be dropped in the channel but nonetheless be corrupted through the effect of fast fading (or other sources of noise and interference). This byte-level corruption was modelled by the second Gilbert-Elliott model, with the same parameters (applied at the byte level) as that of the packet-level model except that p_B (now probability of byte loss) was increased to 0.165. Effectively, this second model emulates fast fading between good and bad conditions. Notice that the packet drops caused by the channel model are not the packet losses resulting from buffer overflow in the congestion experiments.

5.2. Error Resilience

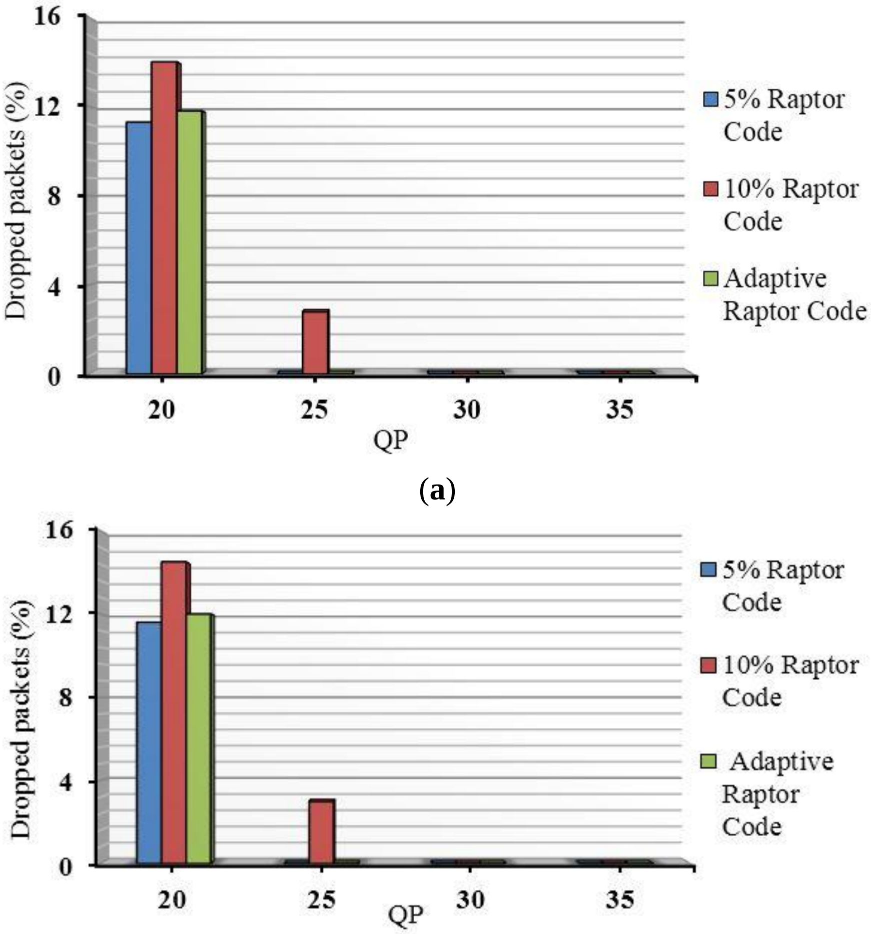
This section concentrates on the impact of the Gilbert-Elliott channel model rather than any packet losses from buffer overflow. As previously outlined, there are two types of channel packet errors: those caused by error bursts of sufficient intensity in which complete packets are dropped and those bursts errors resulting in the corruption of packets. As previously mentioned, in the event of a failure to decode a corrupt packet, a repair is attempted by an ARQ request for additional rateless data if an attempt at decoding fails. If the additional redundant data still fails in allowing successful packet recovery, that packet is then also regarded as dropped. If not, it is regarded as corrupted but suffers from the additional delay required to request redundant data. Upon receiving that additional data, the packet's data is then decoded again.

For this set of experiments, two different video traces were employed for input to ns2 in the WiMAX downlink tests. The first one was the *Paris* sequence (also used in the congestion resilience experiments). The second one was *Football* (high temporal coding complexity) sequence. Both sequences had Variable BitRate (VBR) encoded in Common Intermediate Format (CIF) (352 × 288 pixels/frame) at 30 frame/s.

The RIMR level was set to 5% in the IPPP.... GOP structure (where I is an I-frame followed by all P-frames denoted as PPP…). In these experiments, a single slice per frame was allocated.

The experiments determined the effect of changing the percentage of redundant rateless data or alternatively adopting the adaptive scheme, which initially calculates the amount of redundant data sent through Equation (2). To make a comparison with the adaptive scheme, fixed amounts, 5% or 10%, of redundant data were added to the packets prior to transmission over the WiMAX link.

In Figure 15, it is apparent that increasing the redundancy from 5% to 10% and then changing to the adaptive scheme of Section 4.1 has an effect on packet drops, which impacts video quality, as FEC cannot be applied to dropped packets. However, video content has little impact even though *Football* is much more active than *Paris*. In contrast, QP has a significant impact because the reduction in size of partition C packets as the QP is lowered (refer back to Figure 10) results in a reduction in the number of larger packets.



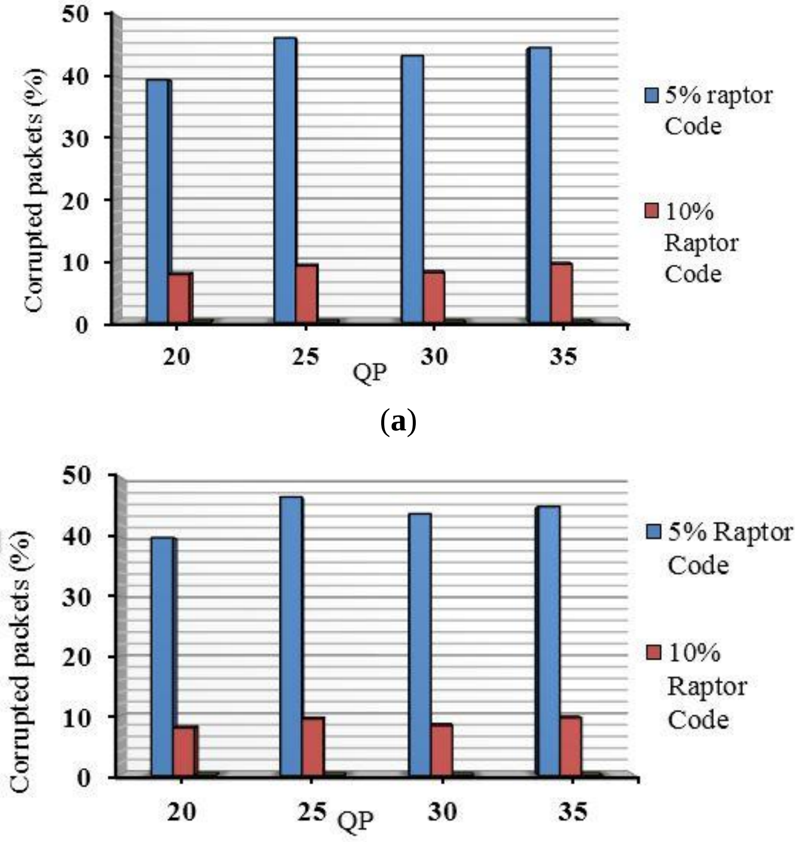
(**a**)

(**b**)

Figure 15. Dropped packets (%) for streaming (**a**) *Football* and (**b**) *Paris* with 5%, 10% and adaptive redundant data.

Comparing the number of corrupted packets in Figure 16, one sees a dramatic increase in their number, with only 5% redundant data but their elimination with adaptive FEC. Thus, for adaptive FEC, the mean delay of corrupted packets (the mean delay including retransmission of all corrupted packets) is zero because there are no corrupted packets. However, in Figure 17 the reconstructed objective video quality, Peak Signal-to-Noise Ratio (PSNR), for 5% and adaptive redundant data is much the same. There is, of course, a penalty from not adapting the redundancy rate according to the channel conditions, because the total delay in retransmitting the corrupted packets (Figure 18), would build up if only 5% rateless coding was applied, leading to larger start-up buffers and discouraging interactive video applications. The higher percentage of dropped packets at low QP (high broadcast quality) and larger packet sizes leads to unacceptable video quality, with the redundant 10% data leading to a drop in quality at QP = 25.

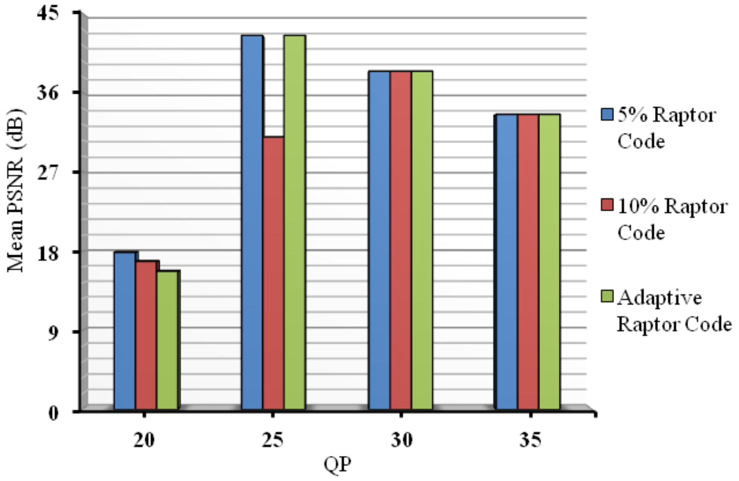
There are small variations in the mean end-to-end packet delays across the QPs (Figures 18 and 19), whose variations are influenced by packet sizes and their propagation times. Therefore, adaptive rateless coding reduces delay considerably over using 5% redundant data but achieves better video quality than when using 10% redundant data. Naturally, delay is an important issue in a real-time service, as it governs buffering provision, enables interactive applications, and if not checked can lead to freeze frame effects.



(**a**)

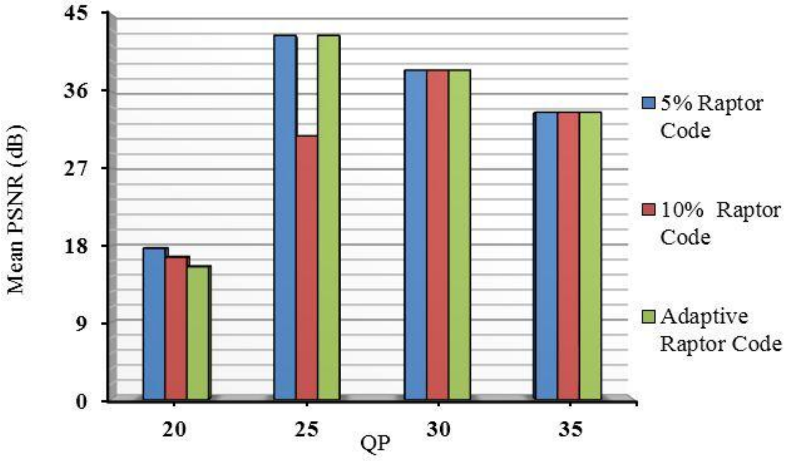
(**b**)

Figure 16. Corrupted packets (%) for streaming (**a**) Football and (**b**) Paris with 5%, 10% and adaptive redundant data.



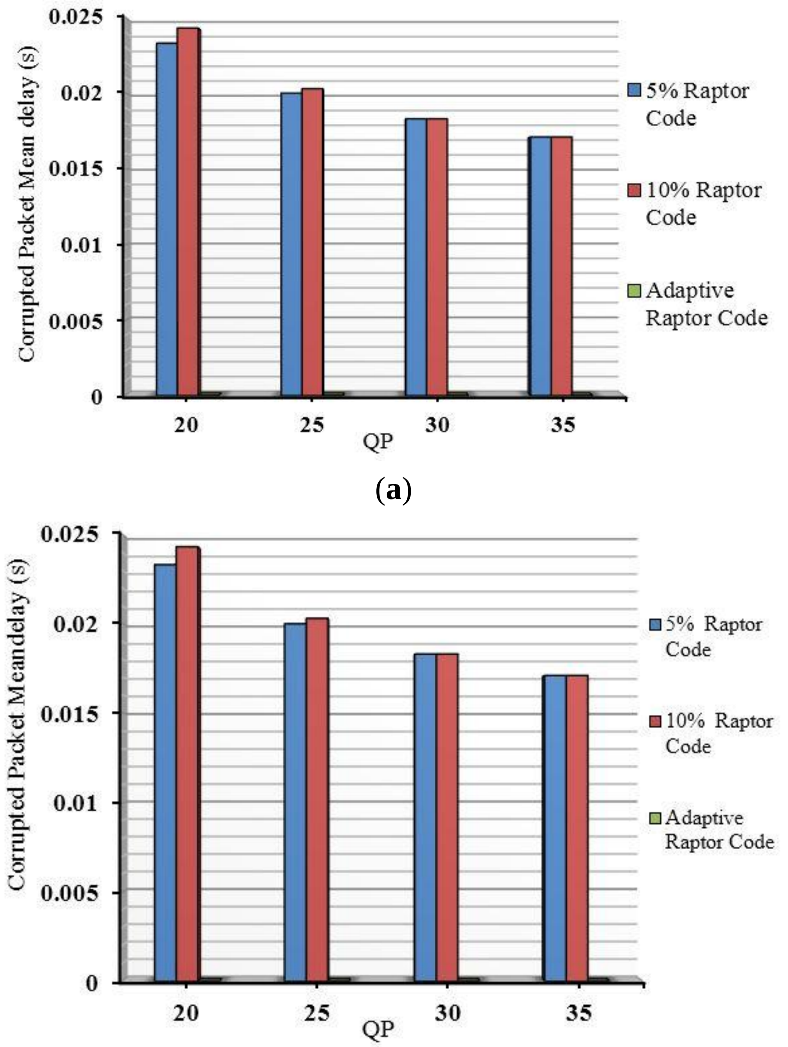
(**a**)

Figure 17. *Cont.*



(**b**)

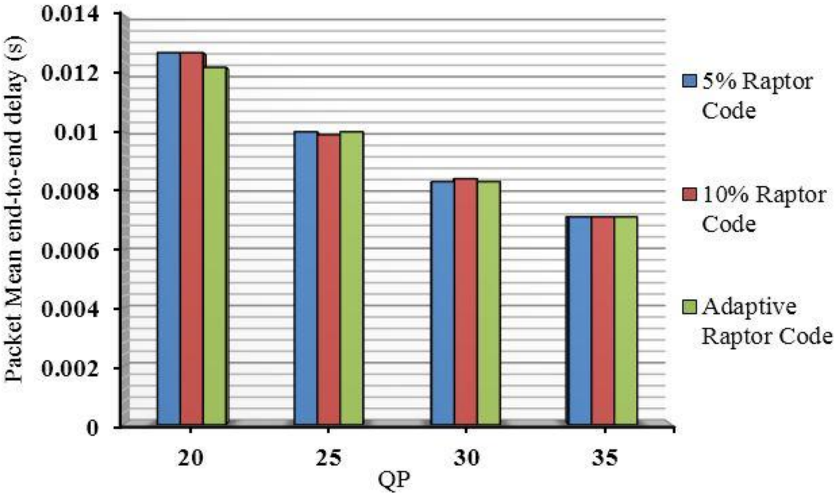
Figure 17. Mean Peak Signal-to-Noise Ratio (PSNR) (dB) for streaming (**a**) *Football* and (**b**) *Paris*, with 5%, 10% and adaptive redundant data.



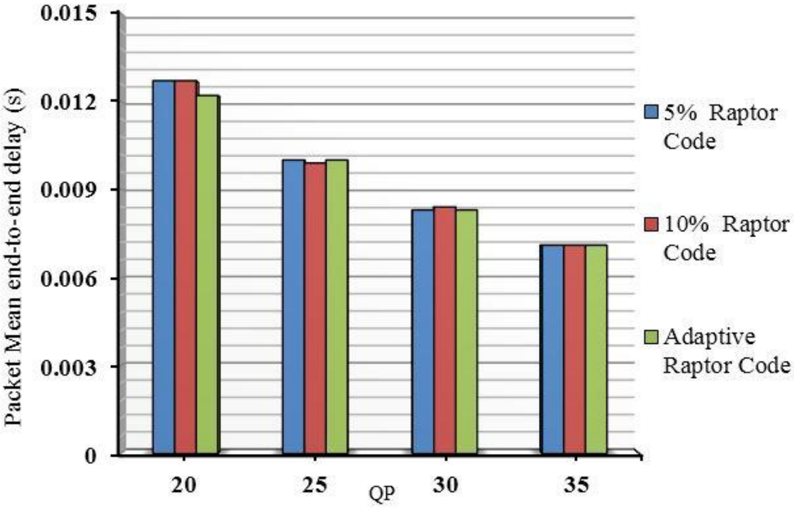
(**a**)

(**b**)

Figure 18. Corrupted packet mean delay (s) for streaming (**a**) *Football* and (**b**) *Paris,* with 5%, 10% and adaptive redundant data.



(**a**)



(**b**)

Figure 19. Packet mean delay (s) for streaming (**a**) *Football* and (**b**) *Paris* with 5%, 10% and adaptive redundant data.

5.3. Congestion Resilience

Figures 20 and 21 show the packet size distribution as a result of coding the two test video clips, *Paris* and *Stefan*, with CIF at 30 frame/s. The sequences were this time coded with a CBR target of 500 kbps. In this distribution, 5.6% intra-coded MBs were randomly added to each frame, equivalent to one row of MBs in CIF. (In the Joint Model (JM) reference software of H.264/AVC used, this results in a gradual decoding refresh within 18 frames, because there is no replication of previously allocated random MB positions in successive frames.) From Figure 20 for *Paris*, about 63% of packets are moderately sized, with 11% relatively large. The remaining 26% are small. The average absolute length of motion vectors in the *Stefan* sequence is 5.5, while for *Paris*, it is only 2.2, implying that there is much more motion activity in *Stefan*. Partially as a result, the more active *Stefan* sequence has fewer smaller packets (Figure 21), than *Paris* at this bitrate.

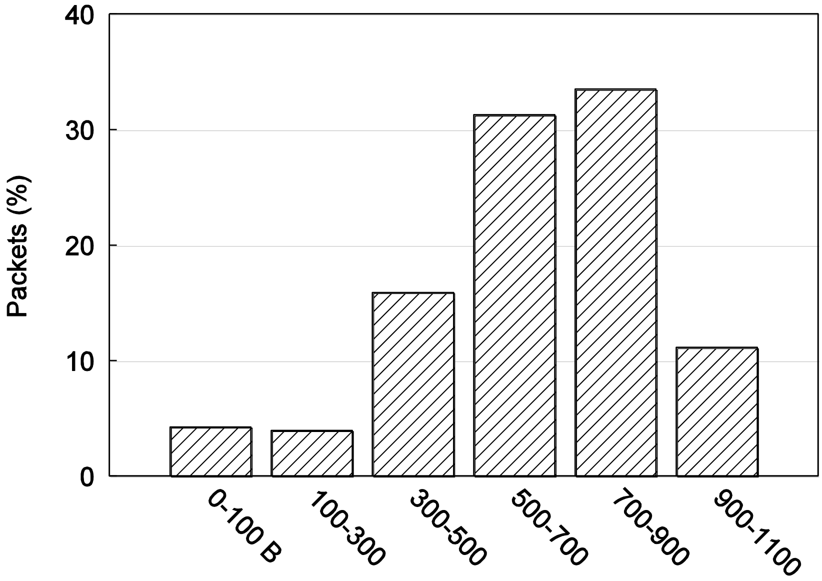


Figure 20. Packet size (bytes) distribution for Paris CIF@30 Hz, coded at a Constant Bit-Rate (CBR) of 500 kbps, with 5.6% Random Intra Macroblock Refresh (RIMR).

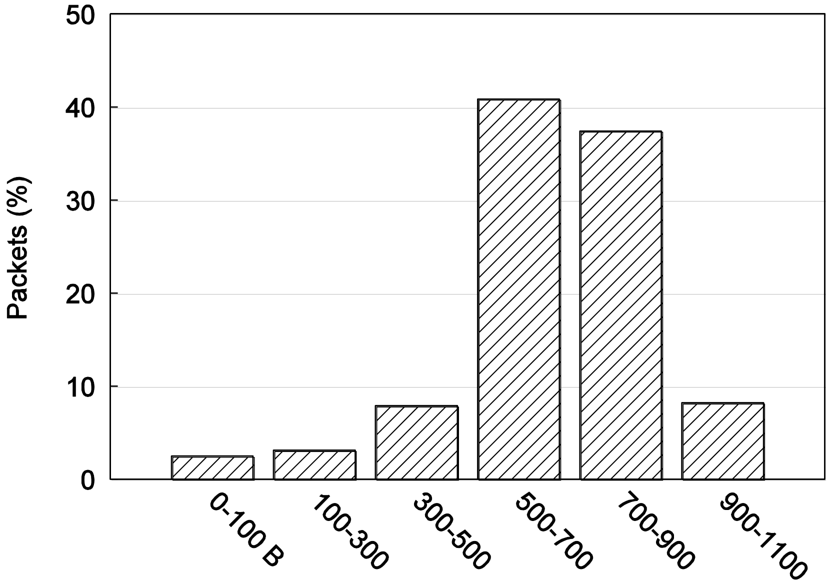


Figure 21. Packet size (bytes) distribution for *Stefan* CIF@30 Hz, coded at a CBR of 500 kbps, with 5.6% RIMR.

Figures 22 and 23 present the result of streaming *Paris* and *Stefan* respectively across the WiMAX link, before and prior to turning on the congesting sources mentioned at about frame number 125. During this period, approximately 2% of the CBR stream packets are dropped through buffer overflow. From the figures, it is apparent that there is a content-dependent effect, as the more active sequence, *Stefan*, does not benefit from the scheme as much as *Paris* does. The reason for this is apparent from looking back at the packet size histograms of Figures 20 and 21. As *Stefan's* packet sizes are predominantly in the range 500–900 B, they are more likely to pose a risk of overflow at the WiMAX BS output buffer. However, for both types of content there is a positive gain from turning on the packet scheduling and, in the case of *Paris*, there is a very definite gain in video quality, which remains stable.

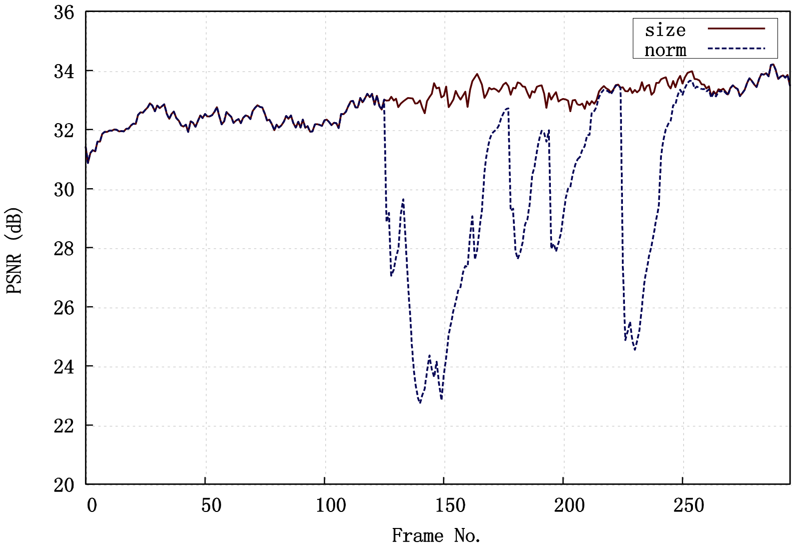


Figure 22. Objective video quality for *Paris* when using both scheduling approaches, by "size" and the equisped "norm".

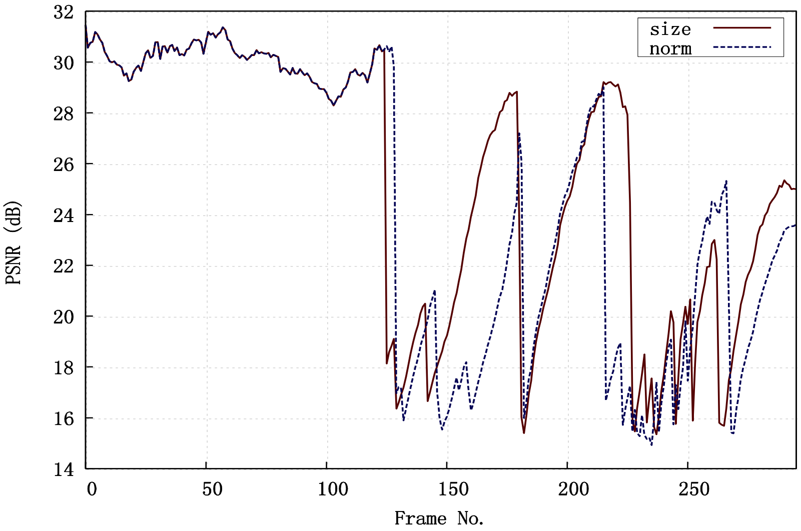


Figure 23. Objective video quality for *Stefan* when using both scheduling approaches, by "size" and the equi-spaced "norm".

6. Conclusions and Discussion of Future Work

Because of the way research is commissioned, it is sometimes not apparent that wireless links suffer not just from an increased level of errors but that the problems of congestion at the input to the wireless channel does not go away either. The errors tend not to be isolated but are correlated as bursts. This paper proposes a scheme that is both error resilient and congestion resilient. The error resilience scheme is multi-faceted as it includes adaptive rateless coding, ARQ error control, and source-coded data partitioning.

The proposed sized-based scheduling solution, specialized to data-partitioned video streams, will certainly benefit the types of content favored by mobile viewers according to quality-of-experience studies, which are studio scenes with limited motion activity. Active sports clips, especially those with small balls, are less attractive but may become more so as the move to higher resolutions on mobile devices continues (e.g., full Video Graphics Array (VGA) format (640 × 480 pixels/frame at 30 Hz) for streaming in Apple's FaceTime). However, in that case the problem of packet scheduling regimes may become more acute. Future work will require a robust investigation of appropriate packet scheduling for congested broadband links for a range of content genres.

High Performance Video Coding (HEVC) [62] emerged as a successor to H.264/AVC during the period of this research. Broadly, HEVC refines and extends the structure of H.264/AVC rather than radically departing from it. However, very few error-resilience tools were included in HEVC, in part because some of these, such as Flexible Macroblock Ordering, were reportedly [63] rarely used in commercial applications. However, this situation provides an opportunity for researchers themselves to add data partitioning to the HEVC source code. Another likely reason why few error resilience tools were built into HEVC is that a full implementation is likely to be computationally demanding, including data partitioning coming at a cost of some other feature of the code. In contrast, direct and efficient implementation of Raptor code has become easier to accomplish with the development of RaptorQ [64], a systematic version of Raptor code software. However, inclusion of such code in a commercial product is subject to a patent in many jurisdictions even if it may be available for educational and research purposes. Such issues do not fall within the scope of this article, though in terms of future research, increased opportunities of confirming the error-correcting performance of Raptor codes now exist. The inner LT code of Raptor is a late decoding type of coding scheme, as sufficient symbols need to be collected before decoding can commence. If network coding is permitted, Switched Code [65] is then a low-latency form of rateless code that is worthy of investigation by us for multimedia applications, as it permits early decoding of packets. Another area for future work is in error resilience. For example, intra-refresh provision in [66] is varied according to an estimate of the rate-distortion trade-off that takes into account expected errors over a wireless channel.

Author Contributions

Laith Al-Jobouri and Ismail A. Ali were chiefly responsible for the experimental contributions in this paper, whereas Martin Fleury contributed the theoretical development and was responsible for the presentation of this paper. Mohammed Ghanbari supervised the overall direction of this research. All authors have read an approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Oyman, O.; Foerster, J.; Tcha, Y.J.; Lee, S.C. Towards enhanced mobile video services over WiMAX and LTE. *IEEE Commun. Mag.* **2010**, *48*, 68–76.
2. Liang, Y.J.; Apostolopoulos, J.G.; Girod, B. Analysis of packet loss for compressed video: Effect of burst losses and correlation between error frames. *IEEE Trans. Circuits Syst. Video Technol*. **2008**, *18*, 861–874.

3. Fleury, M.; Moiron, S.; Ghanbari, M. Innovations in video error resilience and concealment. *Recent Pat. Signal Process.* **2011**, *1*, 1–11.
4. Stockhammer, T.; Bystrom, M. H.264/AVC data partitioning for mobile video communication. In Proceedings of the IEEE International Conference on Image Processing, Singapore, 24–27 October 2004; pp. 545–548.
5. Al-Jobouri, L.; Fleury, M.; Ghanbari, M. Protecting H.264/AVC data-partitioned video streams over broadband WiMAX. *Adv. Multimed.* **2012**, doi:10.1155/2012/129517.
6. Ghanbari, M. *Standard Codecs: Image Compression to Advanced Video Coding*; IET: London, UK, 2003.
7. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576.
8. Ferré, P.; Doufexi, A.; Chung-How, J.; Nix, A.R.; Bull, D.R. Robust video transmission over wireless LANs. *IEEE Trans. Veh. Technol.* **2008**, *57*, 2596–2602.
9. Jammeh, E.; Fleury, M.; Ghanbari, M. Smoothing transcoded MPEG-1 video streams for Internet transmission. *IEEE Proc. Vis. Image Signal Process.* **2004**, *151*, 298–306.
10. Al-Jobouri, L.; Fleury, M.; Ghanbari, M. Engineering wireless broadband access to IPTV. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1493–1506.
11. Declercq, D.; Fossorier, M.; Biglieri, E. *Channel Coding: Theory, Algorithms, and Applications*; Academic Press: Boston, MA, USA, 2014.
12. Wicker, S.B. *Error Control Systems*; Prentice-Hall: Engelwood Cliffs, NJ, USA, 1995.
13. Lin, S.; Costello, D.J. *Error Control Coding*, 2nd ed.; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2004.
14. Girod, B.; Färber, N. Wireless video. In *Compressed Video over Networks*; Sun, M.T., Reibman, A.R., Eds.; Marcel Dekker: New York, NY, USA, 2001; pp. 465–512.
15. Khansari, M.; Jalali, A.; Dubois, E. Low bit-rate video transmission over fading channels. For wireless microcellular system. *IEEE Trans. Circuits Syst. Video Technol.* **1996**, *6*, 1–11.
16. Larzon, L.A.; Degermark, M.; Pink, S.; Jonsson, L.E.; Fairhurst, G. The Lightweight User Datagram Protocol (UDP-Lite), RFC 3828. Available online: http://www.hjp.at/doc/rfc/rfc3828.html (accessed on 17 April 2015).
17. Andrews, J.G.; Ghosh, A.; Muhamed, R. *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*; Prentice Hall: Upper Saddle River, NJ, USA, 2007.
18. Heron, A.; MacDonald, N. Video transmission over a radio link using H.261 and DECT. In Proceedings of the International Conference on Image Processing and Its Applications, Maastricht, The Netherland, 7–9 April 1992; pp. 621–624.
19. Hsu, C.; Ortega, A.; Khansari, M. Rate control for robust video transmission over burst-error channels. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 756–773.
20. Joe, I. An adaptive hybrid ARQ scheme with concatenated FEC codes for wireless ATM. In Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking, Budapest, Hungary, 26–30 September 1997; pp. 131–138.
21. Kallel, S.; Haccoun, D. Generalized Type II Hybrid ARQ scheme using punctured convolutional coding. *IEEE Trans. Commun.* **1990**, *38*, 1938–1946.

22. Lettieri, P.; Fragouli, C.; Srivastava, M.B. Low power error control for wireless links. In Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking, Budapest, Hungary, 26–30 September 1997; pp. 139–150.
23. Shokorallahi, A. Raptor codes. *IEEE Trans. Inf. Theory* **2006**, *52*, 2551–2567.
24. Byers, J.; Luby, M.; Mitzenbacher, M.; Rege, A. A Digital Fountain approach to reliable distribution of bulk data. In Proceedings of the ACM SIGCOMM, Seattle, WA, USA, 2–4 September 1998; pp. 56–67.
25. Jenkăc, H.; Hagenauer, J.; Mayer, T. The Turbo-Fountain. *Eur. Trans. Telecomm. (ETT)* **2006**, *17*, 337–346.
26. Luby, M. LT codes. In Proceedings of 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 271–280.
27. Palanki, R.; Yedidia, J.S. Rateless codes on noisy channels. In Proceedings of the IEEE International Symposium of Information Theory, Chicago, IL, USA, 27 June–2 July 2004.
28. Nguyen, T. An Investigation of Coding Complexity and Coding Rate Performance of Raptor Codes. Master's Thesis, University of Concordia, Montreal, QC, Canada, 2011.
29. Bonello, N.; Chen, S.; Hanzo, L. Myths and realities of rateless coding. *IEEE Commun. Mag.* **2011**, *49*, 143–151.
30. Afzal, J.; Stockhammer, T.; Gasiba, T.; Xui, W. Video streaming over MBMS: A system design approach. *J. Multimed.* **2006**, *1*, 25–35.
31. Jenkăc, H.; Mayer, T.; Stockhammer, T.; Xu, W. Soft decoding of LT-codes for wireless broadcast. In Proceedings of the IST Mobile Summit, Dresden, Germany, 19–23 June 2005.
32. Karp, R.; Luby, M.; Shokorallahi, A. Finite length analysis of LT codes. In Proceedings of IEEE International Symposium of Information Theory, Chicago, IL, USA, 27 June–2 July 2004.
33. Di, C.; Proietti, D.; Telatar, E.; Richardson, T.; Urbanke, R. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inf. Theory* **2002**, *48*, 1570–1579.
34. Luby, M.; Gasiba, T.; Stockhammer, T.; Watson, M. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Trans. Broadcast.* **2007**, *53*, 235–246.
35. Luby, M.; Stockhammer, T.; Watson, M. Application layer FEC in IPTV services. *IEEE Commun. Mag.* **2008**, *46*, 95–101.
36. Hepsaydir, E.; Witvoet, E.; Binucci, N.; Jadhav, S. Enhanced MBMS in UMTS networks and Raptor codes. In Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, Athens, Greece, 3–7 September 2007; pp. 1–5.
37. Rouzbeh, R.; Fleury, M.; Altaf, M.; Sammak, H.; Ghanbari, M. H.264 video streaming with data partitioning and growth codes. In Proceedings of the IEEE International Conference on Image Processing, Cairo, Egypt, 7–10 November 2009; pp. 909–912.
38. Nazir, S.; Stanković, V.; Vukobratović, D. Unequal error protection for data partitioned H.264/AVC video streaming with raptor and random linear codes for DVB-H networks. In Proceedings of the IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11–15 July 2011; pp. 1–6.
39. Nazir, S.; Stanković, V.; Vukobratović, D. Scalable broadcasting of sliced H.264/AVC over DVB-H network. In Proceedings of the IEEE International Conference on Networks, Singapore, 14–16 December 2011; pp. 36–40.

40. Nazir, S.; Vukobratović, D.; Stanković, V.; Andonović, I.; Nybom, K.; Grönroos, S. Unequal error protection for data partitioned H.264/AVC video broadcasting. *Multimed. Tools Appl.* **2014**, doi:10.1007/s11042-014-1883-8.
41. Sgardoni, V.; Bull, D.R.; Nix, A.R. Spectrum efficient cross-layer adaptation of Raptor codes for video multicasting over mobile broadband networks. *Pervasive Mob. Comput.* **2014**, in press.
42. Sgardoni, V.; Nix, A.R. Raptor code-aware link adaptation for spectrally efficient unicast video streaming over mobile broadband networks. *IEEE Trans. Mob. Comput.* **2015**, *14*, 401–415.
43. Li, Y.; Markopoulou, A.; Apostolopoulos, J.; Bambos, N. Content-aware playout and packet scheduling for video streaming over wireless links. *IEEE Trans. Multimed.* **2008**, *10*, 885–895.
44. Tian, D.; Li, X.; Al-Regib, L.G.; Altunbasak, B.; Jackson, J.R. Optimal packet scheduling for wireless video streaming with error-prone feedback. In Proceedings of the IEEE Wireless Communications and Networking Conference, Atlanta, GA, USA, 21–25 March 2004; pp. 1287–1292.
45. Kang, S.H.; Zakhor, A. Packet scheduling algorithm for wireless video streaming. In Proceedings of the International Packet Video Workshop, Pittsburgh, PA, USA, 24–26 April 2002.
46. Chen, C.M.; Lin, C.W.; Chen, Y.C. Packet scheduling for video streaming over wireless with content-aware packet retry limit. In Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing, Victoria, Canada, 3–6 October 2006; pp. 409–414.
47. Liebl, G.; Kalman, M.; Girod, B. Deadline-aware scheduling for wireless video streaming. In Proceedings of the IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6–8 July 2005.
48. Capozzi, F.; Piro, G.; Grieco, L.A.; Boggia, G.; Camarda, P. Downlink packet scheduling in LTE cellular networks: Key design issues and a survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 678–700.
49. Piro, G.; Grieco, L.A.; Boggia, G.; Fortunat, R.; Camarda, P. Two-level downlink scheduling for real-time multimedia services in LTE networks. *IEEE Trans. Multimed.* **2011**, *13*, 1052–1065.
50. Nuaymi, L. *WiMAX: Technology for Broadband Wireless Access*; John Wiley & Sons: Chichester, UK, 2007.
51. Ahmadi, S. *Mobile WiMAX: A Systems Approach to Understanding IEEE 802.16 m Radio Access Technology*; Academic Press: Burlington, MA, USA, 2011.
52. *IEEE Std., 802.16-2009, IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Broadband Wireless Access Systems*; IEEE Computer Society: New York, NY, USA, 2009.
53. Fleury, M.; Ali, I.A.; Ghanbari, M. Video intra coding for compression and error resilience: A review. *Recent Pat. Signal Process.* **2014**, *4*, 32–43.
54. Dhondt, Y.; Mys, S.; Vermeirsch, K.; van de Walle, R. Constrained Inter Prediction: Removing dependencies between different data partitions. In Proceedings of the Advanced Concepts for Intelligent Visual Systems, Delft, The Netherlands, 28–31 August 2007; pp. 720–731.
55. Bing, B. *HD and 3D Broadband Video Networking*; Artech House: Norwood, MA, USA, 2010.
56. Neves, P.; Sargneto, S.; Pentikousis, K.; Fontes, F.; WiMAX cross-layer system for next generation heterogeneous environments. In *WiMAX, New Developments*; Dalal, U.D.D., Kosta, Y.P., Eds.; In-Tech: Vukovar, Croatia, 2009; pp. 357–376.
57. Issariwakul, T.; Hossain, E. *An Introduction to the ns-2 Simulator*, 2nd ed.; Springer Verlag: Berlin, Germany, 2012.

58. Tsai, F.C.D.; Chen, J.; Chang, C.W.; Lien, W.J.; Hung, C.H.; Sum, J.H. The design and implementation of WiMAX module for ns-2 simulator. In Proceedings of the Workshop on ns2: The IP Network Simulator, Pisa, Italy, 10 October 2006.
59. Al-Jobouri, L.; Fleury, M.; Al-Majeed, S.S.; Ghanbari, M. Effective video transport over WiMAX with data partitioning and rateless coding. In Proceedings of the 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), Bradford, UK, 29 June–1 July 2010; pp. 767–772.
60. Jiao, Ch.; Schwiebert, L.; Xu, B. On modeling the packet error statistics in bursty channels. In Proceedings of the IEEE Conference on Local Computer Networks, Tampa, FL, USA, 6–8 November 2002; pp. 534–541.
61. Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005.
62. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1648–1667.
63. Schierl, T.; Hannuksela, M.M.; Wang, Y.K.; Wenger, S. System layer integration of High Efficiency Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1871–1884.
64. Luby, M.; Shokrollahi, A.; Watson, M.; Stockhammer, T.; Minder, L. RaptorQ forward error correction scheme for object delivery. RFC 6330. Available online: http://www.rfc-editor.org/info/rfc6330 (accessed on 17 April 2015).
65. Kadi, N.; Al Agha, K. New degree distribution to improve LT-code in network coding for broadcasting in ad-hoc wireless networks. In Proceedings of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications, Istanbul, Turkey, 26–29 September 2010; pp. 1853–1857.
66. Zhao, P.; Liu, Y.; Liu, J.; Ci, S.; Yao, R. SSIM-based error-resilient rate-distortion optimization of H.264/AVC video coding for wireless streaming. *Signal Process. Image Commun.* **2014**, *20*, 303–315.