A Review of Error Resilience Techniques in Video Streaming

A. Panayides, M. S. Pattichis, C. S. Pattichis, A. Pitsillides

Abstract— **Delivering video data of satisfactory quality over unreliable networks - such as the internet or wireless networks - is a demanding area which has received significant attention of the research community over the past few years. Given the fact that packet loss is inevitable and therefore the presence of errors granted, the effort is directed towards limiting the effect of these errors. A number of techniques have been developed to address this issue. This paper aims to summarize the most significant approaches for: error resilience, error concealment and joint encoder-decoder error control techniques, and to provide a thorough discussion of the benefits and drawbacks of these error control methods. Furthermore, two case studies of error resilience utilization are presented, namely Ad-hoc networks and Multimedia Broadcast Multiple Services (MBMS).**

I. Introduction

Video streaming presents a number of challenges associated with delivering efficient, high-quality video to the end user. Frames parting a video sequence undergo heavy source compression in order to achieve an adequate size suitable for transmission over today's best effort packet based networks. This is made possible due to the considerable amount of temporal and spatial correlation among video frames. The presence of a single bit error however, may cause the entire packet undecodable, clearly not a desirable outcome.

Consequently, designing compression algorithms and the compressed bit stream in such a way that is resilient to errors is the prudent way to address this issue. There are three major approaches which have been developed to tackle error control, each categorized according to where error control actually takes place:

- error resilience at the encoder.
- error concealment is triggered and carried out by the decoder.
- channel adaptive techniques that are based on encoder-decoder interaction via a feedback channel.

The papers by Wang et al. [1], [3] and Villasenor et al. [2] provide a thorough review of the work carried out on error resilience techniques up to 2000. This work covers video coding standards MPEG-1 [4], MPEG-2 (also known as H.262) [5], and MPEG-4 [6], developed by ISO/IEC Moving Picture Experts Group (MPEG) and video coding standards H.261 [7], and H.263 [8], developed by (ITU-T), Video Coding Experts Group (VCEG).

A. Panayides, C. S. Pattichis, and A. Pitsillides are with the Department of Computer Science, University of Cyprus, Kallipoleos 75, P.O. Box 20537, CY 1678 Nicosia, CYPRUS, (csp5ap1@cs.ucy.ac.ucy, pattichi@ucy.ac.cy, Andreas.Pitsillides@ucy.ac.cy).

M. S. Pattichis is with the Department of Electrical and Computer Engineering, Room 229-A, ECE Building, The University of New Mexico, Albuquerque, NM 87131-1356, USA, (pattichis@ece.unm.edu).

In addition, [9]-[15] present a comprehensive synopsis of the H.264/AVC (Advanced Video Coding) [16] and the incorporated error resilience tools used therein. H.264/AVC was jointly developed by the ISO/IEC MPEG and ITU-T VCEG experts, who formed the Joint Video Team (JVT).

The objective of this paper is to provide an outline of the entirety of the error resilience techniques incorporated by video coding standards today, focusing on their implementation in wireless networks.

The rest of the paper is organized as follows. Section II introduces the fundamental concepts of video streaming and a brief description of the incorporated protocols. Sections III, IV and V deal with error resilience. More specifically, section III describes error resilient techniques, section IV presents the error concealment methods and section V incorporates the joint encoder-decoder error control techniques. Section VI provides a case study of error resilience in Ad-hoc and Multimedia Broadcast Multicast Services (MBMS) [17] wireless networks. Finally, section VII provides some concluding remarks.

II. Video Streaming

Video streaming is the delivery of a video sequence over a network in real time, where video decoding starts before the entire video has been transmitted. That is, the client plays the incoming video stream in real time as the data are received. However, we should distinguish between live and on demand video streaming. Essentially, we have two different delivery methods.

The term live streaming is used when various events such as concerts, football games, news, lectures etc. are streamed over the network. To attain this, specific broadcast software and equipment must be employed. This kind of equipment is typically composed by a video camera which records a live source passed as an input to the broadcasting software, which in turn encodes the live source in real time and delivers the resulting stream to a server. The server then forwards the stream to the clients. A client can join a live streaming at any point, and all clients receive the same stream throughout, regardless the time of connection.

On-demand streaming on the other hand tackles archived videos such as movies or weather reports, where the client initiates the stream from the beginning individually. As a result, no client can join the stream after it has been initialized. No specific software is required for on-demand streaming.

It was evident ever since video streaming started gaining wide acceptance and interest that the current internet protocols would be unable to cope with time constraints imposed by real time video delivery. Transmission control protocol (TCP) [18] is highly efficient for Hypertext

Transfer Protocol (HTTP) [19] applications. TCP uses retransmission and traffic monitoring to secure packet delivery to destination. However, this approach is not applicable for video streaming since retransmission time is simply unacceptable in most cases. Given the fact that a limited number of packet losses are tolerable in video streaming and video decoding may recover from such losses, TCP's no loss tolerance simply introduces additional jitter and skew. Furthermore, retransmission may result in alternations of temporal relations between audio and video.

The User datagram protocol (UDP) [20] on the other hand, does not provide any error handling or congestion control, allowing therefore packets to drop out. For this particular reason it is preferred by most commercial streaming software. A significant drawback however, is that UDP information is blocked by firewalls in many networks.

The design of a new internet protocol that would enhance the existing protocols while being suitable for real time data delivery was more than essential. The Real time protocol (RTP) [21] provides end-to-end delivery services for data with real time characteristics such as interactive video and audio. Despite being able to provide real time data delivery, RTP itself does not contain any mechanisms to ensure on time delivery. In the contrary it relies on UDP, TCP for doing so (RTP typically runs over UDP). It does provide however the appropriate functionality for carrying real time content such as time-stamping and control mechanisms that enable synchronization of different streams with timing properties. RTP distinguishes data delivery and control mechanisms and consists of basically two parts: the RTP part which carries the real time data and the Real Time Control Protocol (RTCP) part which is responsible for Quality of Service (QoS) monitoring and extracting information regarding the participants in an RTP session. This information can be later used to improve QoS as it can be supplied as feedback for the encoder to adapt to network conditions, as we discuss in Section VI. It is worth noticing that RTP/UDP/IP, RTCP/UDP/IP packets are sent over distinct ports. Two important features introduced by RTP is the use of mixers and translators. It is often the case that not all participants in an RTP session have the same connection capabilities as far as bandwidth is concerned. In such cases a mixer is used to transform an incoming higher bandwidth stream to a lower one, reflecting a participant's capabilities. Translators are triggered to surpass delivery limitations imposed when a client is behind a firewall. By installing two translators, one at each side (source and receiver), and through a secure connection, translators resolve the problem raised by a firewall of blocking the incoming stream outside the internal network.

RTP payload contains the real time data being transferred while the RTP header contains information characterising the payload such as timestamp, sequence number, source, size and encoding scheme. As we have already mentioned, RTP packets are usually transferred over UDP, which in turn are encapsulated in IP packets, hence UDP/IP headers.

III. Error Resilience

Error resilient techniques aim to encode the compressed bit stream in such a way that the transmission errors' impact upon decoding and reconstruction of the video data will be minimal. To achieve this, the encoder must add redundancy to the compressed bit steam. Redundancy bits are additional to data bits and are the ones responsible for improved quality in the presence of transmission errors. The involved mechanism employed of course does not come at no cost, as encoders adding redundancy become less efficient than normal encoders. However, in the long run, the benefit is obvious. The problem is then focused on maximizing error resilience with the smallest possible amount of redundancy bits.

Error resilient video coding techniques can be subdivided in the following approaches: Robust Entropy Coding, Data Partitioning, Flexible Macroblock Ordering, Arbitrary Slice Ordering, SI/SP synchronization/switching pictures, Incorrect State and Error Propagation, Unequal Error Protection with Layered Coding, and Multiple Description Coding, which are also summarized in Table 1. The rest of this section deals with these techniques.

A. Robust Entropy Coding

A compressed video stream's high vulnerability to transmission errors emerges in part from the fact that a video coder employs non-resilient variable length coding (VLC) to represent various symbols. Consequently, once a bit error occurs or a bit is lost, immediately the involved and subsequent codeword(s) are constituted non-decodable (see Fig. 1), since the decoder is not able to match the appropriate bits to the appropriate parameters.

Fig. 1. An example of the occurrence of an error in a bitsream. The bits are properly decoded until the occurrence of an error (indicated by X) makes the remaining bits non-decodable.

One should note that fixed length coding (FLC) is not susceptible to this problem since the knowledge of the beginning and the end of each codeword limits the loss to that single codeword. However, FLC's provide poor compression efficiency and therefore are not considered.

1) Re-synchronization markers

Re-synchronization markers [3] constitute a simple, yet efficient way to address the bit stream synchronization problem. Such markers may be placed strategically in the bit stream (MPEG-1/2, H.261/3) or periodically (MPEG 4). Placing re-synchronization markers strategically (every fixed number of blocks, variable number of bits) suffers from the increased probability that active bit stream areas may be corrupted. Conversely, periodically placing the markers (variable number of blocks, fixed number of bits) reduces this probability while it simplifies the search of resynchronization markers and supports network packetization. Therefore re-synchronization markers are best placed periodically. A marker should be designed in such a way that it will notably differ from other code words, concatenations of code words and minor permutations of code words.

The end of a marker should follow some kind of header information incorporating spatial and temporal locations or other in-picture predictive information concerning the subsequent bits. The decoder can then resume decoding properly after the interruption which occurred by the presence of the re-synchronization marker.

The frequency and length of re-synchronization markers can be thought of redundancy-wise. That is, the more frequent and longer the markers are, the more redundancy bits are employed and conversely. In addition, the presence of a marker typically interrupts in-picture prediction mechanisms, such as motion vector (MV) or discrete cosine transform (DCT) coefficient prediction, contributing in this way in increasing the added redundancy. However, longer and frequent re-synchronization markers enable the decoder to recover faster from transmission errors and therefore restrain their effect by reaching synchronization quicker. Consequently, in practical video coding systems, long and frequent markers are employed.

Fig. 2. An example of re-synchronization marker utilization. After an error has occurred, the decoder jumps to the next re-synchronization marker and continues decoding properly.

2) Reverse variable length coding (RVLC)

While conventional VLC's only enable forward unique decoding, reverse variable length codes RVLC's [22] also provide unique backward decoding. The underlying idea is not to waste the information represented by the bits that intervene between the corrupted bits and the next re-synchronization marker when we can make use of them; So, instead of jumping to the next re-synchronization marker and continue decoding onwards, we jump to the next synchronization marker and start decoding backwards to decode the non corrupted bits.

Fig. 3. An example of RVLCs utilization. After an error has occurred, the decoder jumps to the next re-synchronization marker and starts decoding both backwards, marked as recovered data in Fig. 3, and also forwards.

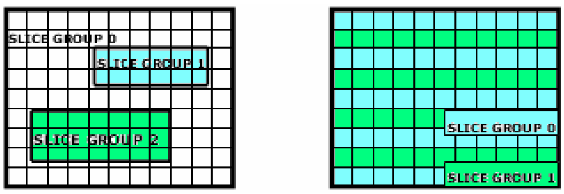
The introduced complexity by RVLC is not prohibitive in terms of coding efficiency, contrary to what was originally believed. RVLC can be designed with near perfect entropy coding efficiency, and thus efficient implementation. Nevertheless, knowledge gained through practical use has shown that RVLC may as well prove to be more efficient than VLC for certain applications, providing greater error resilience. RVLC has been adopted by both MPEG-4 and H.263 in conjunction with insertion of synchronization markers. Design principles, classification and coding efficiency are further discussed in [23]-[33], in classic papers by Villasenor et al. [23] - [25], [28] – [29], Tsai et al. [26] – [27], and in more recent ones [30] - [33].

B. *Data Partitioning*

The basic idea in data partitioning lies in the observation that not all bits in a bitstream carry equal information. On the contrary, data bits can be categorized according to their importance, with certain bits being more important than others. Data partitioning in H.264/AVC allows the partitioning of a normal slice in up to three parts and each part is paired accordingly with unequal error protection (UEP, II.D) during transmission. Data partition (DP) A contains the most important slice information such as MB types and MVs, and possible loss or corruption of DP A, constitutes the remaining two partitions of no use. Second in importance comes DP B, which consists of intra-coded block patterns (CBPs) and I-block transform coefficients, while DP C incorporates inter CBPs and P-block coefficients. More detailed description can be found in [9]-[10] along with recommended actions when partition loss is detected [9].

C. *Flexible Macroblock Ordering (FMO)*

An innovative error resilient feature introduced by H.264/AVC is flexible macroblock ordering [34] – [35]. FMO is essentially a slice structuring approach, where a frame is parted into independently transmitted and decoded slices. Prediction between slices is not allowed and consequently corrupted packets do not propagate error to subsequent packets. FMO aims to evenly distribute errors throughout a frame. In this manner and in conjunction with proper utilization of the spatial relationships between slices and MBs therein, concealment of errors becomes much more efficient. Fig. 4 depicts FMO utilization. It is the case however that a packet carrying a whole slice is dropped. To enhance robustness in such cases, H.264/AVC allows the transmission of *redundant slices* (RS) (a redundant slice being a slice describing the same MBs in a bitstream). RS can be coded in a different manner with respect to the primary slices (i.e. different coding parameters) and are utilized in the absence of a clear primary slice.



(a) (b)

Fig. 4. Partition of a frame into slices using Flexible Macroblock Ordering (FMO). Two different partitions are given in (a) and (b).

D. Arbitrary Slice Ordering (ASO)

Arbitrary Slice Ordering (ASO) ([10], [11], [15]) enables slices to be essentially transmitted independently of their order within a picture. As a result, they can be also decoded out of sequence, reducing thus the decoding delay at the decoder. ASO is particularly effective in environments where out-of-order delivery of a packet is possible such as the internet or wireless networks, or packet based networks in general.

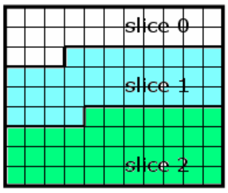


Fig. 5: Partition of a frame into slices using Arbitrary Slice Ordering (ASO).

E. SP/SI synchronization/switching pictures

SP/SI [36], [37] are two new picture types introduced in the H.264/AVC design that allow the decoder to switch between two or more pre-encoded bitstreams, constructed from the same source sequence, but of different bandwidth and quality. Beside the obvious benefit of switching between two bitstreams, this dual nature feature proves particularly efficient in terms of error resilience, especially in the presence of a feedback channel which enables the decoder to trigger the encoder to perform a bitstream switch, regaining in that way lost synchronization resulting from data losses or errors. Nevertheless, valuable bandwidth is preserved, since recovery from an error does not incorporate the transmission of an I-frame. The SP/SI scheme can be used for operations such as fast-forward, reverse, etc.

F. Incorrect state and error propagation

The compression and encoding procedure's philosophy in video coding standards remains basically the same since the introduction of MPEG-2. However, enhancements in subsequent standards and recently in H.264/AVC have managed to improve coding efficiency significantly.

Briefly looking into the design of video coding standards, we observe that pictures belonging to a sequence are first split into blocks, and then coded and transmitted. Subsequent pictures are next reconstructed at the decoder using prediction from previously decoded ones along with the encoder's supplied information for improved efficiency. Essentially two modes of coding exist:

- *Intra-mode:* Intra-mode is the procedure where intra-prediction is used for coding a video frame. That is, all the information used for coding originate from the picture itself and block samples are predicted using spatially neighbouring samples of previously coded blocks. Typically the first picture of a sequence is coded in intra-mode and following intra-coded pictures depend on available bitrate and imposed time constraints.
- *Inter-mode:* Inter-mode is the procedure where inter-prediction is used for coding a video frame. When intra-mode is not employed, inter-mode is used, and is further subdivided into the following distinct modes:
  - *P-mode*: P-mode uses prediction from previously decoded frames. Contrary to intra-mode where all information are derived from the picture itself, in inter-mode, the encoder's side provides all the necessary information for accurate motion estimation of the spatial displacement between the decoder's reference picture and the current picture in the sequence at the encoder. This procedure is described as motion compensation.
  - *B-mode*: Whereas in P-mode at most one motion compensated signal is employed, B-mode provides the ability to make use of two motion compensated signals for the prediction of a picture. B-mode is also referred to as bi-prediction as not only it allows the utilization of previously decoded pictures but also the utilization of forthcoming ones.

The presence of an error originates the incorrect reconstruction of the frame (state) at the decoder. Once the decoder is found in a different state than the encoder, subsequent frames reconstructions referencing this incorrect state will also result in error. This situation can produce significant error propagation. To address this issue, some form of re-initialization must take place in the prediction loop in order to limit and/or prevent error propagation.

The exclusive use of I-Frames in the computation of forward frames eliminates error propagation.

The insertion of intra-coded macroblocks (MBs) has proved to be a considerable and effective technique towards the limitation of error propagation. In addition, the complexity introduced complicates the encoder. Intra-coding however, involves higher bit rate than inter-coding. Consequently, the use of intra-coding is limited and should be employed wisely. Several approaches have been proposed throughout the years but none significantly outperforms some other [1] – [3]:

- Periodic intra-coding of all MBs, intra-codes different MBs in each frame in some predefined order so that after a certain number of frames all MBs have been intra-coded at least once.
- Preemptive intra-coding is based on previous knowledge of a channel loss model, which allows the estimation of which MBs are most vulnerable to errors. Likewise, this approach places the intra-coded MBs in areas of highest activity.
- Random placement has also proved to perform quite well.

Intelligent *intra-block refreshing by Rate Distortion (RD)* is a technique that selects a block coding scheme which minimizes a certain cost function (adopted by the

H.264/AVC [38], [39]). In addition, *multiple* (two or more) *reference picture motion compensation* mode is allowed, which enables the utilization of more than one reference frames during the prediction phase (more detailed descriptions of both can be found in [9] – [15] and references therein).

It is obvious that channel characteristics directly impact the number of intra-coded MBs used. Channel knowledge may be obtained by a number of ways, one of which is point to point communication with back channel, which we consider later in section V. Feedback based – channel adaptive paradigms are examined in VI.

G. Unequal Error Protection (UEP) with Layered Coding (LC)

Layered Coding (LC) codes a video using a base layer and one to many enhancement layers. The base layer provides limited but of satisfactory quality video, while each of the enhancement layers is used to incrementally improve the decoded video quality. In order for LC to be constituted an efficient error resilience tool, it must be paired with UEP, so that the base layer is protected more strongly. To attain this, enhanced forward error correction (FEC) [40] and automatic repeat request (ARQ) schemes are employed [1] – [3]. Corruptions occurring in enhancement layers are not of equal importance with corruptions emerging in the base layer coding of video. Typically, the base layer includes vital information, loss of which may result in having no video at all. The LC structure and philosophy supports users of different bandwidth capacities and decoder capabilities to access the same video data at varying qualities (scalable coding).

H. Multiple Description Coding

Multiple description coding (MDC) codes a data source into a number of descriptions, correlated and of roughly equal importance. That is, the source sequence is coded into multiple bitstreams with minor differences, which are in turn transmitted independently. The underlying idea is that decoding a single intact stream provides adequate quality, while the decoding of multiple robust streams provides higher quality. The correlation among descriptions is the characteristic that enables the decoder to tell when a description is corrupted or not and thus provide a satisfactory quality level out of every description. MDC provides a reliable sub channel, even in error prone networks. A plethora of MDC approaches exist and are summarised in [3], whereas a more recent and comprehensive review can be found in [39].

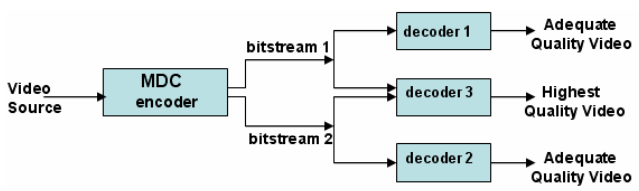


Fig. 6. An example of multiple description video coding.

TABLE I
ERROR RESILIENCE TECHNIQUES

Technique		Video Coding Standards	Channel Adaptive Technique
Robust Entropy Coding	Resync Markers	MPEG-2/ H.263	NO
	RVLC	MPEG-4/ H.263	NO
Data Partitioning		MPEG-2/ H.263	NO
FMO		H.264/AVC	NO*
ASO		H.264/AVC	NO*
Redundant Slices		H.263	NO*
SP/SI		H.264/AVC	NO*
Incorrect state and Error Propagation	Periodic I-MB	MPEG-4/ H.263	NO*
	Preemptive I-Coding	MPEG-4/ H.263	NO*
	Random I-Coding	MPEG-4/ H.263	NO*
	Intra-block refreshing by RD	H.264/AVC	NO*
Multiple Reference		MPEG-4/ H.263	NO*
UEP & LC		MPEG-4/ H.263	YES
MDC		MPEG-4/ H.263	YES

*Error resilience technique can be used both in a non channel adaptive and a channel adaptive environment.

IV. ERROR CONCEALMENT

It is very likely that transmission errors will result in loss of information. Error concealment techniques estimate and replace the missing data in an attempt to conceal errors in the decoded stream. Error concealment uses spatial or temporal interpolation based on correctly decoded data, in an attempt to recover lost data.

Spatial interpolation, temporal interpolation and Motion compensated temporal interpolation techniques are discussed in this section ([1] – [3], [9] – [15], and references therein).

A. Spatial interpolation

A simple method used for recovering corrupted data is to use spatial interpolation. Video image intensity at a single pixel is spatially interpolated from correctly decoded image intensity at surrounding pixels. However, due to the fact that all blocks or MBs of the same row are usually placed in the same packet, the only available adjacent blocks are those of the rows above and below, not a representative sample of the damaged pixels in most cases. As a result, only the boundary pixels in neighbouring blocks are used for interpolation. Even in this approach, correctly recovering missing pixel values is extremely difficult. Instead, the DC (average) value is estimated and used to replace every corrupted pixel. Computing the DC value is done very efficiently. A better approach is to use an interleaved packetization mechanism so that the loss of each packet only affects every other block or MB.

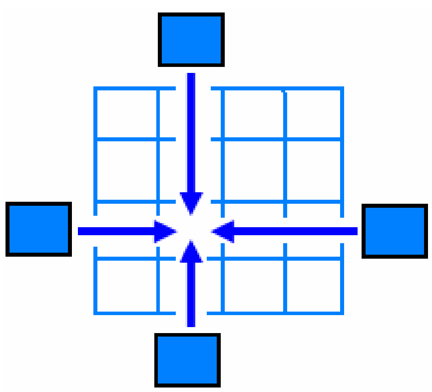


Fig. 7: Spatial Interpolation. The corrupted region is recovered from surrounding correctly decoded pixels.

B. Temporal interpolation

Another approach to recover a corrupted MB is motion compensation (MC) temporal prediction, which copies the pixel values from the same spatial location of the previous frame (freeze frame) (see Fig. 8). This approach is effective when there is no motion involved but is susceptible to problems in the presence of motion. To address potential problems, the MV is also used in deciding the pixel location from the previous frame. MC temporal techniques generally provide better results than spatial interpolation techniques. Combination of both techniques works better in the estimation of missing MVs.

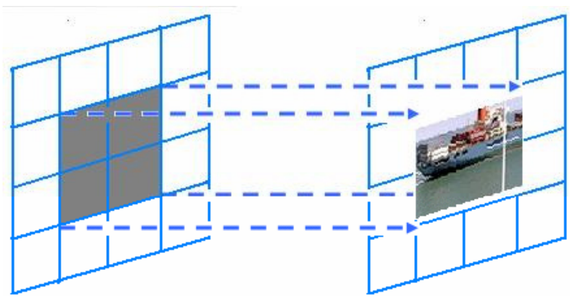


Fig. 8. Temporal Interpolation.

C. Motion compensated temporal interpolation

MV is the corner stone for a plethora of the video standards since inter-frames computation is based upon the knowledge of the MV in combination with the DCT coefficients of the prediction error. Consequently, corruption of the MV significantly reduces the decoded video quality prior to the decoding of an intra-frame. To achieve quality error concealment at the decoder, data partitioning may have to be employed by the encoder. As described above, data partitioning packs important data such as the MV and employed mode and transmits them with increased error protection, thus limiting the probability that this data will be damaged or lost. This mode is employed by both MPEG-4 and H.263. However, there is high probability that an error may occur, and therefore actions must be taken to deal with them. In such a case, both the coding mode and the MVs need to be estimated. A simple way to compute the coding mode is to assume that the MB is coded in the intra-mode and therefore use only spatial interpolation to recover the affected blocks. Another is to derive the damaged MB by collecting statistics of the surrounding MBs and select the most likely MB. For MV estimation several approaches exist which include among others: (a) assume the lost MVs to be zeros, (b) use the MVs of the corresponding block in the previous frame, (c) use the average of the MVs from spatially adjacent blocks, (d) use the median of MVs from the spatially adjacent blocks, and (e) re-estimate the MVs [3]. Typically, when a MB is damaged, its horizontally adjacent MBs are also damaged, and hence the average or mean is taken over the MVs above and below.

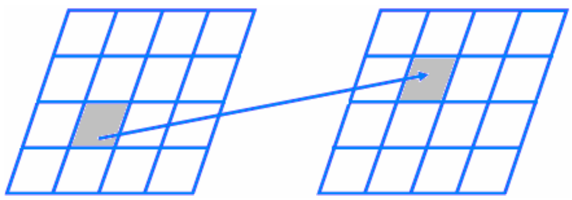


Fig. 9. Motion compensated temporal interpolation. The motion vector in conjunction with the transmitted DCT coefficients are used to achieve effective prediction of missing or corrupted blocks as motion compensated blocks are based on a previously decoded frame.

TABLE 2
ERROR CONCEALMENT TECHNIQUES

Technique	Video Coding Standards	Channel Adaptive Technique
Spatial Interpolation	MPEG-1/ H.261	NO
Temporal Interpolation	MPEG-1/ H.261	NO
Motion Compensated Temporal Interpolation	MPEG-4/ H.263	NO

V. JOINT ENCODER-DECODER ERROR CONTROL

Instead of acting independently, the encoder and the decoder can set up a communication channel which can be then used to tackle error control more effectively. The decoder can send feedback to the encoder regarding the lost information, such as the position of the corrupted data, and the encoder can then adapt to these conditions to limit decoding errors. A comprehensive review of feedback based error control can be found in [42].

This section encompasses two such approaches which are described below: error tracking based on feedback information and choosing which frame to use based on feedback information [3], [42].

Fig. 10. Encoded-Decoder communication with back channel.

A. Error tracking based on feedback information

Upon notification by the decoder that an error has occurred, the encoder can:

- Reinitialize prediction using an I-frame. Simple and relatively straightforward approach, however it suffers from the fact that it employs higher bit rate for intra-coding. Clearly, this is not an attractive option.
- Avoid using the error affected region in the prediction of subsequent frames.
- Employ the same error concealment technique used by the decoder for the affected frames. In this way, the encoder's and decoder's reference frames will match when coding the next frame.

The first two techniques require that the encoder be informed of the spatial extend of the decoded error, whereas the last one incorporates the duplication of the decoder error concealment procedure for the damaged frames. All techniques help in reducing the propagated error before the insertion of the next I-frame.

B. Choosing which frame to use based on feedback information

An alternative approach is to use feedback from the decoder to the encoder to specify the location of the occurred error, and then use this information to determine a new reference frame. That is, instead of using the most recent reference frame for coding the next frame, use an older reference frame which is available at the decoder and known to be error-free. For the utilization of this method, the encoder and the decoder need to store multiple previously coded and decoded frames. Note that this does not necessarily add delay at the decoder. Nevertheless, when compared to using an I-frame for coding the next frame, this approach is significantly more efficient when recently encoded/decoded frames are used. In this approach, the encoder decides how the reference frame is chosen.

This approach is employed by both MPEG-4 and H.263 in what is called NewPred and Reference Picture Selection (RPS) respectively. Two modes of operations which can be found in the MPREG-4 specification are ACK and NACK:

- In the ACK mode, the encoder only encounters acknowledged (i.e. correctly received) frames for prediction. By doing so, it minimizes error propagation but is susceptible to the use of relatively "old" (further apart) frames, which leads to poor compression. This happens because acknowledgements may arrive late at the encoder.
- On the other hand, NACK mode uses the last coded frame as reference for prediction, unless negative acknowledgement is received. In this way, the most recent coded frames are used. However, in the event of an error with delayed negative acknowledgement, error propagation increases.

The effectiveness of these feedback-based approaches depends on the round trip delay (RTD). In general, effectiveness decreases as RTD increases constituting this approach inapplicable for applications such as broadcast, multicast or pre-encoded video. Conversely, these approaches are better suited for video teleconferencing applications.

In [9], [10], [13], [14] applications and constraints are discussed in more detail. Essentially three distinct applications are identified, the nature of which determines the constraints and the appropriate protocol environment [14]. Conferencing applications typically have less than 1 Mbps of bandwidth and are the most demanding, demanding less than 100ms end-to-end transmission delay. Pre-encoded video streams available for download on the other hand operate between 1-8 Mbps and are not really susceptible to delay constraints. Lastly, streaming applications, which operate at 50kbps-1.5 Mbps, fall between conference and pre-encoded applications [10], [14].

TABLE 3
JOINT ENCODER-DECODER ERROR CONTROL TECHNIQUES

Technique	Video Coding Standards	Channel Adaptive Technique
Error Tracking based on Feedback Information	MPEG-4/ H.263	YES
Choosing which frame to use based on feedback information	MPEG-4/ H.263	YES

VI. ERROR RESILIENCE IN TODAY'S WIRELESS NETWORKS

The discussed error resilience techniques are applicable on whole or in part to the entirety of today's wireless networks. In [13], [14], [43], there are discussions on the use of H.264/AVC features for applications in wireless environments. In this section, we examine two case studies of recent advances in wireless networking: ad-hoc networks and multimedia broadcast multicast services (MBMS). The nature of ad-hoc networks is such that, error-resilience approaches are most appropriate. Multimedia Broadcast Multicast Services (MBMS) is a currently developed standard which has already attracted the majority of the network providers, making video streaming QoS an indispensable target.

TABLE 4
ERROR RESILIENCE IN AD-HOC NETWORKS AND MBMS

Error Resilience	Ad-hoc	MBMS
Non-Channel Adaptive	Intra-Update FMO Multiple Frames MDMC	Resync Markers RVLC Header retransmission
Channel Adaptive	RPS	IBR
	LC with ARQ	

A. Error resilience in Ad-hoc networks

Contrary to the classic wireless networks (GSM [44], UMTS [45]) which are based on infrastructure and base stations connected to a wired backbone network, ad-hoc networks do not require any network infrastructure. Mobile ad-hoc networks or MANETs are a collection of geographically distributed mobile nodes that interact "on the move" with one another over a wireless medium. The nature of such networks suggests that mobile nodes parting a MANET will carry out user tasks such as application traffic, while at the same time, they will be responsible for network control and routing protocols. This exact dual operation in conjunction with the rapidly changing topology (and therefore connectivity) and limited battery life is what constitutes reliable video transport more challenging than over other wireless networks. Connection paths are highly error prone while at the same time the endurance of an existing path cannot be guaranteed due to frequent node failures. However, multiple paths can be established between sources and destinations, offering a useful tool when designing video coding and transport schemes.

1) Non-channel adaptive error resilience

In the absence of a feedback channel, the most commonly used methods to stop error propagation are intra-update, flexible macroblock ordering (FMO), the use of multiple reference frames, and multiple description motion compensation (MDMC).

A plethora of *intra-updating* schemes exist, all of which are effective and provide increased video quality when employed:

- Experimental results verify that smaller intra-frame periods perform better in scenarios with higher packet loss when intra-coding a picture (I-frame).
- Random intra-macroblock updating presents behaviour similar to periodic insertion of intra-coded pictures. One should notice however that sequences with more movement benefit more from frequent intra-updates.
- Intra-updating a whole macroblock line randomly is also an efficient approach when intra-updating.

In general, random intra-macroblock updating performs better than picture and random macroblock line intra-coding [46], [47]. Moreover, when compared to the original no updating video sequence, random intra-macroblock updating may improve the signal to noise ration (SNR) up to 5db [46].

For ad-hoc networks, the most appropriate measure for evaluating error resilience is in the presence of burst errors. Experimental results [20] show the boosting effect of *FMO* when the burst error is limited within a single frame. In the case however that a single burst error spreads to consecutive frames, it is better to switch to another error resilience technique, since FMO is mostly suitable for errors occurring within a single frame. Random intra-macroblock updates is such a technique.

Contrary to what was believed prior to simulations, results for random error scenarios have shown that the use of *multiple reference frames* is not always synonymous with better compression or superior error resilience. Despite a slight gain in bitrate, the use of multiple frames turned out to be inefficient as the use of a single reference frame provided better results, besides the obvious gains in terms of reduced memory requirements at the encoder and the decoder [47].

MDMC is built on top of block based motion compensated prediction (MCP). MDMC uses linear superposition of two predictions from two previously decoded frames. Even and odd motion vectors for frames are sent on separate paths and studies show increased quality when applying MDMC. MDMC provides adequate error resilience and is mostly efficient when channels have low error rate [48].

2) Channel adaptive error resilience

The incorporated channel adaptive error resilience techniques discussed are feedback based reference picture selection (RPS) and layered coding (LC) with selective automatic repeat request (ARQ).

In the *feedback based RPS* scheme, even frames are sent on one path, and odd frames are sent on another. Negative feedback (NACK) is sent for lost packets and positive feedback (ACK) for correctly received ones. Paths are marked as "good" and "bad" accordingly. The nearest possible correctly decoded frame is used as the reference picture, which is either a frame sent on a good path that has not been acknowledged yet, or an acknowledged one.

The RPS scheme does not introduce any decoding delay. In the event that a delay of a certain level is acceptable, *layered coding* with *selective ARQ* can be used. In this case, the video stream is layer coded, with base layer packets and enhancement layer packets transmitted on different paths. The receiver sends an ARQ if the base layer is lost, which is then retransmitted on the path of the enhancement layer. Its error resilience does not depend on the round trip time (RTT) but the decoding and display delay are determined by RTT.

Both channel adaptive techniques provide considerable error resilience and whether a delay is acceptable or not, are used accordingly. They both outperform MDMC and make effective use of the possibility offered by ad-hoc networks of setting multiple paths between source and destination [49], [48].

B. Error resilience in MBMS

Multimedia Broadcast Multicast Services (MBMS) [17] is a new standard being developed by 3rd Generation Partnership Project (3GPP) [50] to enable a new class of spectrum efficient multimedia services. Delivering video of satisfactory quality over MBMS is challenging due to mobility, diversity of signal conditions, low power and spectrum utilization requirements of the receivers [51]. Video error resilience in MBMS services is critical to maintain consistent quality for end users. However, conventional error resilience techniques for IP multicast are not applicable to MBMS. Consequently, new techniques must be adopted or already existing ones adjusted accordingly. Interactive techniques where lost packets are retransmitted following a request by the decoders (receivers) are ruled out in MBMS systems since such actions may lead to feedback implosion. Automatic repeat request (ARQ) and forward error correction (FEC) strategies also cannot guarantee the wireless transmission within multicasting time constraints.

Error localization techniques providing error resilience in MBMS include *resync markers*, *RVLCs*, and *header retransmission*. Header retransmission is significant during a multicasting session, since it contains the information required for a user to join the session at any time.

Adaptive group based intrablock refresh (IBR) technique is largely employed in MBMS systems. In MBMS and multicasting scenarios, using RTCP feedback reports, the average signal strength conditions of the group members can be determined [51], [52]. Based on that, the percentage of encoded intra-blocks can be determined. If the receivers are experiencing high error rates, the encoded intra-block percentage is increased and vice versa. Essentially, a partitioning of multicast users into groups is performed, aiming to maximize video quality of the entire group.

VII. CONCLUSION

Growing demand for Quality of Service (QoS) over today's unreliable packet based networks has motivated the development of a number of error resilient techniques, most of which have already been adopted in recent coding standards.

In this paper, we described the fundamental concepts of video streaming and the incorporated protocols. We reviewed the most significant error resilient approaches, in an attempt to describe a generic framework through which error control is achieved. We discussed encoder-based techniques that provide a compressed bitstream that is resilient to errors, while minimizing the added bitstream redundancy. We also examined a number of error-concealment techniques that the decoder uses for reconstructing a corrupted video frame. We also reviewed some of the latest error control techniques based on encoder-decoder communication, where the encoder regulates its operations according to feedback send by the decoder. Despite their obvious effectiveness, these latter approaches suffer from delay constraints, making them inapplicable for a wide range of applications such as in many video conferencing applications. Lastly, we discussed error resilience in ad-hoc networks and MBMS, to underline the extensive deployment of error control in video streaming over wireless networks.

We conclude that error control techniques have been studied thoroughly over the past years, and this is evident by the fact that the most widely used coding standards, MPEG-x and H.26x, and recently H.264/AVC, include a plethora of error control tools. With video streaming applications continually increasing and wireless networking gaining increasing popularity, error control techniques are likely to continue to grow into the future.

REFERENCES

[1] Y. Wang, Q. F. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, Vol. 86, No. 5, pp. 974-97, May 1998.

[2] J. D. Villasenor, Y.-Q. Zhang, J. Wen, "Robust video coding algorithms and systems," *Proc. IEEE*, Vol. 87, No. 10, pp. 1724-33, Oct. 1999.

[3] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Proc. Mag.*, vol. 17, pp. 61-82, July 2000.

[4] ISO/IEC JTC 1, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video," ISO/IEC 11172 (MPEG-1), Nov. 1993.

[5] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), Nov. 1994.

[6] ISO/IEC JTC1, "Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 visual version 1), April 1999; Amendment 1 (version 2), February, 2000;Amendment 4 (streaming profile), Jan., 2001.

[7] ITU-T, Video Codec for Audiovisual Services at px64 kbit/s, ITU-T Recommendation H.261, Version 1: Nov. 1990; Version 2: Mar. 1993.

[8] ITU-T, "Video coding for low bit rate communication," ITUT Recommendation H.263; version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.

[9] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, “Error Resiliency Schemes in H.264/AVC Standard,” Elsevier J. of Visual Communication & Image Representation (Special issue on Emerging H.264/AVC Video Coding Standard), Vol. 17(2), April 2006.
[10] T. Wiegand, G. J. Sullivan, G. Bjψntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst.Video Technol.*, vol. 13, pp. 560–576, July 2003.
[11] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi, “Video coding with H.264/AVC: tools, performance and complexity,” *IEEE Cir. Syst. forVideo Technol. Mag.*, Vol. 4(1), pp.7-28, 2004.
[12] N. Ozbek and T. Tunali, “A Survey on the H.264/AVC Standard, ” Turk J Elec Engin, vol. 13, no. 3. 2005.
[13] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, “H.264/AVC in wireless environments,” *IEEETrans. Circuits Syst. Video Technol.*, vol.13, pp. 657–673, July 2003.
[14] S. Wenger, “H.264/AVC over IP,” *IEEE Trans. Cir. Syst. Video Technol.*, vol. 13, pp. 645–656, July 2003.
[15] G. Sullivan, P. Topiwala, and A. Luthra, “The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions,” In Proceedings of SPIE annual meeting 2004: Signal and Image Processing and Sensors, volume 5558, pages 454--474, Denver, 8 2004.
[16] Joint Video Team of ITU-T and ISO/IEC JTC 1, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC),” Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, March 2003.
[17] 3GPP Technical Specification 3GPP TR 29.846: “Multimedia Broadcast/Multicast Services.”
[18] J. Postel, “Transmission Control Protocol,” RFC 793, 1981.
[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext Transfer Protocol-HTTP/1.1,” RFC 2616, 1999.
[20] J. Postel, “User Datagram Protocol,” RFC 768, 1980.
[21] RFC 1889, RTP: A Transport Protocol for Real-Time Applications. Jan 1996.
[22] Y. Takishima, M. Wada, and H. Murakami, “Reversible variable length codes,” *IEEE Trans. Commun.*, vol. 43, pp. 158–162, Feb.–Apr. 1995.
[23] J. Wen and J. D. Villasenor, “A class of reversible variable length codes for robust image and video coding,” in *Proc. 1997 IEEE Int. Conf. Image Processing (ICIP)*, vol. 2, Santa Barbara, CA, pp. 65-68, Oct. 1997.
[24] J. Wen and J. D. Villasenor, “Reversible variable-length codes for robust image and video transmission,” *Proc. 31st Asilomar Conf. on Signals, Systems, and Computers*, vol. 2, pp. 973-9, Nov. 1997.
[25] J. Wen and J. Villasenor, “Reversible variable length codes for efficient and robust image and video coding,” in *Proc. IEEE Data Compression Conf*, pp. 471–480, 1998.
[26] W. Tsai and J. L. Wu, “On constructing the Huffman-code based reversible variable length codes,” *IEEE Trans. Commun.*, vol. 49, pp.1506–1509, Sept. 2001.
[27] C.-W. Tsai and J.-L. Wu, “Modified symmetrical reversible variablelength code and its theoretical bounds,” IEEE Trans. Inf. Theory, vol. 47, no. 6, pp. 2543–2548, Sep. 2001.
[28] K. Lakovic and J. Villasenor, "On design of error-correcting reversible variable length Codes," *IEEE Commun. Letters*, vol. 6(8), pp. 337-339, 2002.
[29] K. Lakovic and J. Villasenor, “Design considerations for efficient reversible variable length codes,” *Signals, Systems and Computers, Conference Record of the Thirty-Sixth Asilomar Conference*, vol.1, pp. 262 – 266, 3-6 Nov. 2002
[30] C. W. Lin and J. L. Wu, “On Error Detection and Error Synchronization of Reversible Variable-Length Codes,” *IEEE Transactions on Communications*, Vol. 53(5), May 2005.
[31] L Xu, S. Kumar, and M.K. Mandal, “Coding efficiency evaluation of RVLC codes,” *Circuits and Systems, 48th Midwest Symposium*, vol. 2, pp. 1553-1556, Aug 2005.
[32] L. Xu and S. Kumar, “Error Resiliency Measure for RVLC Codes,” *IEEE Signal Processing Letters*, vol 13(2), Feb. 2006.
[33] J. Y. Huo, Y. L. Chang,L. H. Ma, and Z. Luo, “On constructing symmetrical reversible variable-length codes independent of the Huffman code,” J Zhejiang Univ SCIENCE A 7(Suppl. I):59-62, Feb 2006.
[34] S. Wenger, and M. Horowitz, “Flexible MB Ordering – A New Error Resilience Tool for IP-Based Video,” in Proc. IWDC 2002, Capri, Italy, Sept. 2002.
[35] S. Wenger, “FMO: Flexible Macroblock Ordering,” ITU-T JVT-C089, May 2002.
[36] M. Karczewicz and R. Kurceren, “A proposal for SP-frames,” ITU-T SG16 Doc. VCEG-L27r1, 2001.
[37] M. Karczewicz and R. Kurηeren, “The SP and SI Frames Design for H.264/AVC”, IEEE Transactions on Circuits and Systems, vol. 13, no. 7, July 2003.
[38] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra. “Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard,” IEEE CSVT, 6(2):182–190, April 1996.
[39] G.J. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression,” IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74-90, Nov. 1998.
[40] Y. Wang, A. R. Reibman, S. Lin, “Multiple Description Coding for Video Delivery,” *Proc. IEEE*, Vol. 93, No. 1, pp. 57-70, January 2005.
[41] J. Rosenberg and H. Schulzrinne, “An RTP Payload Format for Generic Forward Error Correction,” RFC 2733, 1999.
[42] B. Girod and N. Fδrber, “Feedback-based error control for mobile video transmission,” *Proc. IEEE*, vol. 97, pp. 1707–1723, Oct. 1999.
[43] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, “H.264/AVC in Wireless Environments,” *IEEE Transactions on Circuits and Systems*, vol. 13, no. 7, July 2003.
[44] GSM webpage http://www.gsmworld.com
[45] UMTS webpage http://www.umtsworld.com
[46] C. T. Calafate, M. P. Malumbres, and P. Manzoni. “Performance of H.264 Compressed Video Streams over 802.11b Based MANETs,” ICDCSW, pp. 776-781, 24th International Conference on Distributed Computing Systems Workshops - W6: WWAN (ICDCSW'04), 2004.
[47] C. T. Calafate and M. P. Malumbres. “Testing the H.264 Error-Resilience on Wireless Ad-hoc Networks,” 4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications, Zagreb, Croatia, July 2003.
[48] S. Lin, Y. Wang, S. Mao, and S. Panwar, “Video transport over ad-hoc networks using multiple paths,” *in IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 57—60, May 2002.
[49] S. Lin, S. Mao, Y. Wang, and S. Panwar. “A Reference Picture Selection Scheme for Video Transmission over Ad-hoc Networks Using Multiple Paths*,” in Proc .of ICME*, Aug. 2001.
[50] 3GPP webpage http://www.3gpp.org
[51] P. Sanigepalli, K. Hari, and F. Borko, “Error resilient video over multimedia broadcast multicast services (MBMS),” Multimedia Tools and Applications, Volume 28, Issue 1, Jan 2006.
[52] S. Praveenkumar, H. Kalva, and B. Furht, “An Efficient Application of Video Error Resilience Techniques for Mobile Broadcast Multicast Services (MBMS),” *IEEE Workshop on Wireless Multimedia*, Dec 2004.