Application of H.263+ Video Coding Modes in Lossy Packet Network Environments

Jörg Ott[1] · Stephan Wenger[2] · Gerd Knorr[2]

Universität Bremen TZI · Technische Universität Berlin

jo@tzi.uni-bremen.de · {stewe,kraxel}@cs.tu-berlin.de

Abstract: The quality of real-time audio and video information transmitted via today's Internet suffers severely from often significant packet losses. While this problem is well understood and solved for existing audio coding schemes, support from the video coding standards themselves is required for video streams. This paper presents the newly introduced error resilience mechanisms built into the second version of H.263 (1998), known under its working name H.263+, and addresses the corresponding packetization format issues that together significantly improve the image quality at packet loss rates up to 20 per cent. In particular, it is support from the video coding algorithm itself peered with appropriate transport layer mechanisms that leads to significant improvements of perceived image quality for communicative as well as retrieval applications at moderate bit rates up to some 100 kbit/s.

1 Introduction

Motion video constitutes the most important means for conveying rich and descriptive contents: in interactive communications as well as for information retrieval. However, experience with real-time information transmission (audio as well as video) through the wide-area Internet is in general not satisfactory: packet loss renders audio streams unintelligible and severely deteriorates the quality of a video stream at the receiver. While a simple redundancy scheme applied at the real-time transport layer is sufficient to cope with most packet loss scenarios for audio communications [1], a far more complex scheme is needed to adequately solve this problem for video streams.

Much of the coding efficiency of motion video coding schemes stems from the use of inter-picture prediction mechanisms: a predictive or P frame *n* typically refers to the previously coded frame *n-1* and the bit representation of frame *n* contains only the (delta) information needed to construct frame *n* from *n-1*. Obviously, these inter-coding modes only work if the reference frame *n-1* is available (and correct, identical to the reference frame inside the coding loop) at the receiver. Packet losses may invalidate this precondition and therefore the quality of the decoded image typically suffers significantly. Consequently, today inter-coding modes are typically not applied for video communications in the Internet; current practice in the available tools is to primarily (if not exclusively) use intra-coded frames (I frames). This, however, significantly increases the size (in bits) of a coded frame by a factor as high as 7 to 12, so that, at a given bandwidth, the frame rate (and thus the perceived quality of the *motion* video stream) is noticeably reduced.

Inter-coding modes can only be effectively applied if the video coding algorithms provide means for dealing with errors. While the error resilience mechanisms of H.261 — which is used as video coding standard in most of today's tools for interactive communications — can only deal with bit errors (this is due to its history as video coding standard for use on ISDN lines), H.263+ is the first video coding standard to incorporate a variety of error resilience techniques that are effectively applicable to all of today's relevant networks including packet-based networks such as the Internet.

---

[1] Universität Bremen, Technologie-Zentrum Informatik, Digitale Medien und Netze, Bibliothekstr. 1, D-28359 Bremen, Germany, +49 421 218-2085, +49 421 218-7000 (fax)

[2] Technische Universität Berlin, Institut für Kommunikations- und Softwaretechnik, Fachgebiet Kommunikations- und Betriebssysteme, Sekr. FR 6-3, Franklinstr. 28/29, D-10587 Berlin, +49 30 314-73160, +49 30 314-25156 (fax)

These error resilience video coding mechanisms need to be intertwined with transport layer mechanisms and control protocols: applications have to be able to determine certain network characteristics to adapt their coding and transmission strategies (dynamically) to varying network conditions. Also, the range of applicable error resilience strategies that may be employed depends on the type of video application. That is, for a certain network condition and application type, a suitable combination of error resilience modes can be chosen that provides for these circumstances optimal video quality in spite of transmission errors — which can be proven by simulations that do take into account conditions as they can be found in today's Internet.

This paper is organized as follows: in the second section, we outline the central application scenarios that benefit from error resilience mechanisms in H.263+, in section three, we characterize a typical networking scenario as prevalent in today's Internet. Section four provides the necessary background information on H.263+ and the real-time transport and the control protocol used for audiovisual communication in packet-based networks. In section five, we introduce the error resilience modes of H.263+ that are applicable in packet-based networks along with a supporting packetization format. Section six describes the necessary support for those mechanisms required from the real-time transport and control protocols. Section seven derives a set of rules for applications on how to use H.263+ error resilience mechanisms for varying network conditions. Section eight summarizes the results of simulations we have carried out for different network conditions. Section nine concludes this paper with a brief summary and an outlook.

2 Application Types

Video applications can be characterized in several dimensions, three of which are relevant for the discussion in the remainder of this paper. First of all, (not only from a media transmission point of view) *communicative* and *one-way* applications can be distinguished. Communicative applications support communication between (human) users, i.e. humans interact directly with one another, while one-way applications have a human user only at one end of a communication relationship interacting with some (server) computer system at the other. Furthermore, *interactive* and *non-interactive* applications are distinguished. Interactive applications require that the user receives sufficiently timely feedback on her actions, while for non-interactive ones there is no upper bound for feedback (if there is any feedback possible at all); i.e. in interactive distributed applications, the perceived application performance depends on timely information exchange, in non-interactive ones it does not. Those two dimensions are combined for simplicity: communicative interactive applications are referred to as interactive, one-way interactive as semi-interactive, and all others as non-interactive applications. Finally, distributed applications may be categorized according to the number of participating entities into (traditional) point-to-point applications and multipoint applications — the latter of which may again be roughly subdivided into small group multicast applications and large group multicast or broadcast applications.

This spectrum can be constrained in various ways from a practical perspective. First of all, the error resilience mechanisms as described in this paper are of primary importance for interactive applications. Non-interactive applications can in principle treat their information streams as data files and employ reliable data distribution protocols — unless sender and/or receiver cannot store the entire feed, the group size is too large, etc. In this paper, the only non-interactive applications we consider are broadcast applications (with potentially large groups of receivers). Furthermore, semi-interactive applications typically operate in a point-to-point fashion at any given moment (because only a single individual is likely to control the media retrieval even if several ones are receiving the media streams). Finally, truly interactive applications are restricted to point-to-point communications and small groups.

With these assumptions in place, the three following major applications types remain:

a) Interactive applications include point-to-point multimedia communications (video telephony, a1) as well as multipoint conferencing applications with a small number of users (a2). Because of the usually high level of interaction, this application type is very sensitive to delay: some hundred to at most three hundred milliseconds total delay is still acceptable for audio conversations.

b) Retrieval applications belong to the category of semi-interactive applications and comprise video-on-demand style applications and — as a special case — web-based interactions with video servers that provide live or pre-recorded video feeds as part of a web page. This application type in general has a single interacting recipient. In contrast to a), no social protocols need to be conveyed over the video channels and hence immediate feedback becomes less important; however, timing constraints still apply as it is typically the visual feedback that a human uses to adjust controls of the remote server. Assuming that remote video servers will not be controlled in an very fine-grained fashion via the wide area Internet, an upper delay bound of roughly one second is deemed acceptable.

   For considering adaptive video coding algorithms, it is important to note that two types of retrieval applications can be distinguished: those applications that encode the video data stream immediately before sending it (e.g. in case of live video feeds of surveillance systems) and those that only packetize and transmit pre-recorded and pre-encoded video feeds from a mass storage device. While the former applications can adapt their coding strategies to the feedback as discussed in this paper, the latter have no influence on the bit stream they send and hence have virtually no means to adapt — unless the video feed has been recorded several times with different encoding schemes and the sending application chooses the most appropriate coded stream depending on the feedback. For the remainder of this paper, we always assume that type b) applications are capable of influencing the coded bit stream being transmitted over the network.

c) Broadcast applications constitute a third application type here. These are non-interactive applications but as they generally do not apply reliability techniques as described above the use of error resilience mechanisms is appropriate. They can be treated similarly to retrieval applications except that the timing constraints are even more relaxed since there is virtually no interactivity. Hence, latencies of several seconds or even a minute are tolerable.

   Note that the above considerations on pre-recorded bit streams may apply to some broadcast applications as well and we make the same assumptions as above.

3 Network Characteristics

3.1 Relevant Network Parameters

For real-time communications, the parameters of most importance to real-time applications and their underlying protocols include the delay, the (required) throughput, and the experienced packet loss.

3.1.1 Delay

Four different factors affect the delay between capturing an image on the sending and displaying it on the receiving side:

1. Codec delay — the time it takes to generate (encode) the bit "stream" representing the (next) captured frame as well as the time required to decode the bit stream at the receiving end; this includes the time to wait for future frames to be captured if e.g. certain prediction schemes shall be applied; in case of pre-recorded and pre-codec video streams this delay is zero.
2. Packetization delay — the time it takes until a minimum (if any) number of bits is available or all necessary parts of a frame are coded to fill a packet. If forward error correction (FEC) mechanisms (e.g. using exclusive or operation) are applied, an additional FEC delay may be introduced on the receiving side as — in case of a packet loss — the receiver has to wait for further packets to arrive before it is able to reconstruct the lost packet [2, 3]. For the purpose of this paper, we consider a potential FEC delay to be contained in the packetization delay.
3. Transmission delay — the time the network requires to forward a piece of information from the sender to the recipient; this time comprises medium access time (if any), queuing delays in routers, and propagation delay through each link.

4. Playout delay — additional delay artificially introduced at the receiving end to compensate for varying transmission delays (also termed *jitter*) and ensure continuous playback of the information stream.

In this paper, we refer to each of those four delay types by the respective name and refer to the sum of these four factors as *latency*. From the human user's point of view, only the overall latency is noticed regardless of which factors contribute to it.

For interactive communication scenarios, the goal is to minimize the latency — even at the cost of higher bit rate. This means to focus on addressing those delay factors that can be favorably influenced by using the appropriate coding modes as well as additional mechanisms at the sender and receiver. For semi-interactive applications, latency is less important, for typical broadcast application, it is almost negligible.

3.1.2 Throughput Characteristics

Throughput can be expressed by two parameters: the (expected) average bit rate (e.g. in bits per second) and a traffic specification describing the actual characteristics of the information stream that lead to the mean bit rate value. The average bit rate is typically used to describe a network (link) capacity as well as the average transmission rate of an encoder while the traffic specification refers to the shape of a (variable bit rate) data flow injected into a network. A traffic specification may be arbitrarily complex in order to be able to cover the full range of possible data flows (refer e.g. to [4]). However, for practical reasons, we restrict our considerations to the following parameters for a traffic specification:

- inter-packet interval or number of packets per second;
- number of packets required to carry an independently decodable segment (termed a packet sequence); and
- maximum packet size which depends on the maximum transmission unit (MTU) in the network.

The traffic injected by the application — the total bit rate as well as the traffic shape — may influence the observed packet loss in the network.

In this paper, for H.263+ video streams we assume a fixed frame rate and consequently a fixed number of packet sequences per second and a fixed inter packet sequence interval. This is done to allow a better quality comparison by means of spatial quality measurement, like signal-to-noise ratio (SNR) or a more sophisticated mechanisms like PQS [5]. What is expected to vary for a given setting is the quality of a frame, the individual packet size and/or the number of packets in a packet sequence. The simplification above doesn't have any influence to the algorithms, that are applicable also for variable frame rates – as long as those frame-rates justify the term 'motion video' and thus make mechanisms like the reference picture selection mode or even Inter picture prediction useful.

The throughput characteristics are orthogonal to the application type.

3.1.3 Packet Loss

The final parameter of interest is the packet loss rate of a network. Packet loss can be measured as the fraction of packets lost during a video communication session. In addition, the packet loss distribution — e.g. random losses vs. bursty losses — is of relevance to error resilience techniques. Finally, in case of multiparty communications, the correlation of packet losses between individual recipients is relevant for error resilience mechanisms as well.

On one hand, the packet loss rate depends on the overall network load and cannot be influenced by a particular application. On the other hand, expectations and experience — confirmed by simple tests that were carried out by the authors — suggest that applications can impact their experienced packet loss to some degree by shaping their emitted (video) traffic in a network friendly way. For example, in a presumably unloaded and well-provisioned Internet environment, simple tests between TU Berlin, the University of Bremen, the University College London, and the MIT showed that evenly distributed packets as well as packet sequences of two packets experienced loss of at most 10% per packet, but

packet sequences of three packets sometimes showed a packet loss probability of more than 30%. Moderate variations in the total bit rate an application is injecting into the network does not seem to impact the packet loss rate. In our tests, we varied the bit rate by one order of magnitude (from around 8 kbit/s to 120 kbit/s) without observing a correlation with the observed packet loss probability. These observations are in line with other were confirmed by other observations [6].

3.2 Internetworking Environment

In today's Internet (or more general: any IP-based network), access types for endpoints can be roughly grouped into three different categories that are likely to cover most the infrastructure in place. Those categories can be described as follows based upon the aforementioned three transport characteristics.

1. Corporate or campus IP-based intranets with a throughput in the order of magnitude of 1, 10, or 100 Mbit/s, a transmission delay of some 10 milliseconds and a packet loss rate of less than two per cent. This applies to most well-maintained intranets based on LAN technology such as 10-BaseT and 100-BaseT Ethernet or FDDI and to larger enterprises and most members of the research community with sites often linked by 2 Mbit/s leased lines or ATM backbones.
2. IP networks with ISDN speed connections (BRI, fractional T1) and low to medium packet loss rate of no more than 5%, at bit rates between 56 kbit/s and a few 100 kbit/s. This is the typical connectivity scenario for some number of private and for most business users.
3. IP networks with modem speed dialup connection that have a low error (i.e. packet loss) rate due to low bit rates around 30 kbit/s and a medium (serialization) delay — both characteristics only apply as long as the link bit rate is not exceeded. Today, this includes the largest fraction of private Internet users.

As modems are about to become faster (56k modems) and alternative methods for obtaining access to the Internet appear for private users (e.g. cable modems and xDSL technology), in the mid-term, access type 3 converges to access type 2. On the other end of the scale, the typical endpoint in a corporate network will generally not (be allowed to) use more than a similar amount (some 100 kbit/s) per audiovisual communication relationship. Also, and most important, the maximum usable transmission rate is constrained by what is available from the wide area Internet. The typical throughput for video transmissions as used in the Multicast backbone (Mbone) — which then is able to also accommodate a larger number of sessions — is somewhere between 64 and 128 kbit/s, and this is a reasonable rate for unicast communications as well. Also, this rate is roughly what is needed when gatewaying video streams to / from ISDN-based audiovisual equipment.

As the maximum throughput, the observed transmission delay and packet loss probability are also governed by the wide area Internet (i.e. the involved ISPs and their interconnections) rather than by the characteristics of the local access link: transmission delays may be larger than one second and packet loss rates much higher than five per cent.

These observations paired with the fact that H.263+ provides reasonable quality for many of today's video applications found in the Internet form the basis for us to choose the following networking scenario as good match for the wide-area Internet:

- throughput within the range from 64 to 128 kbit/s — as the actually available bandwidth varies dynamically, applications should be able to measure and to adapt to these changing conditions (within certain bounds, of course).
- wide range of packet loss probability virtually 0 to 30 per cent with a random loss distribution and — in case of multicast communications — largely uncorrelated packet losses; and
- transmission delay of several tens to some hundred milliseconds.[3]

[3] In essence, the assumption is that the network delay must small enough to leave room for additional error resilience delay (part of which is codec delay) so that the total latency remains acceptable for a given application type. If the network delay itself is already beyond the acceptable threshold, the respective application type(s) will not be used.

These numbers are roughly confirmed by various measurements carried out on the Internet for unicast and multicast communications. In tests between TU Berlin on one side and the University of Bremen (9 hops), UCL (15 hops), and MIT (22 hops) on the other, a throughput of up 120 kbit/s was always achieved; the average loss rates measured for individual packets were around 3%, around 5%, and some 10%, respectively, and this loss rate did not seem to have any correlation with the throughput rates ranging from 8 kbit/s to 120 kbit/s. Typically, individual packets were lost randomly. The observed round-trip delay was below 50 ms (but sometimes more than 150 ms) for Bremen, between 50 and 60ms for London, and between 150 and 250 ms (though occasionally up to 700 ms) for Boston.

Other measurements confirm our (rather experimental) findings for unicast communications. For communication between INRIA and UCL, for example, Bolot and Vega-García investigated packet loss for 64 kbit/s audio payload: 320 byte packets transmitted at 40 ms intervals during talkspurts. They observed a packet loss of 4% in the early morning and 16% in the afternoon at a random a packet loss distribution: individual packet losses dominate and the probability of consecutive packets being lost decreases geometrically fast away from the origin [7, 8].

While we did not carry out tests for multicast, Handley [6] and Yajnik et al. [9] gathered statistics on Mbone communications. Their results are roughly similar to the findings for unicast communications. The packet loss distribution is also of largely random nature with typically one or two packets being lost in sequence. The observed packet loss also varies more than for unicast: 50% of the receivers reported a mean loss rate of around 10% or lower but 80% had sometimes intervals with more than 20% packet loss. As for unicast, the average packet loss varied depending on the time of day. At transmission rates between 10 kbit/s and 120 kbit/s Handley did not find a correlation between throughput and packet loss. In addition, both studies report that the packet loss experienced by individual receivers in largely uncorrelated. Handley found a probability of significantly less than 10% that a single packet would reach all recipients in multicast sessions with some hundred or more geographically dispersed participants.

4 The Background: Transport and Control Protocols and H.263+

4.1 An Overview of H.263+

In the series of ITU-T recommendations describing the coding of video signals, H.263 is the most innovative one. Currently, a new version of H.263 is about to undergo the final standardization process. This new version will after its final approval by the ITU-T study group 16 (expected January 1998) be known as H.263+ (1998), but is also well known under its working name of H.263+.

The working principle of H.263+ is similar to the ones of the older ITU-T recommendation for video coding H.261 and to the MPEG family of ISO standards. Inter picture prediction, augmented by motion compensation, is used to reduce temporal redundancy by coding only the delta information between older and newer frames. Spatial redundancy is reduced by the means of DCT based transform coding accompanied by Huffman- or arithmetic coding of the resulting coefficients.

Similar to the four optional modes of version 1 of H.263, H.263+ offers 16 optional modes, which can be independently chosen by the coder. However, certain restrictions apply in terms of mode combinations. A non-normative appendix of H.263+ contains guidelines on the recommended mode combinations to allow easier capability negotiation and to achieve higher interoperability levels.

Most of the optional modes, including all four modes which are present in today's H.263, are designed to allow a tradeoff between computational complexity and coding performance/picture quality. Those modes — if used — may improve error resilience by reducing the number of bits of the data stream without effecting the quality. However, three of the new modes are explicitly intended to achieve a high gain in error resilience. These modes are discussed in more detail in section 5.1 below.

H.263+ was designed as the general ITU-T video coding solution for bit rates below 1 Mbit/s. At higher bit rates, H.262 (the ITU's reference to MPEG 2 video [10]) can be used. H.263 was — out of his history — designed for interactive applications, and consequently no coding mechanisms were

introduced as mandatory features (in the baseline) which severely impede the applicability of H.263 for those applications. As typical example for such mechanisms, B-frames or any type of bi-directional prediction can be mentioned; they do improve coding efficiency but also add coding delay. However, to allow efficient support for non delay-critical applications, B-frames were introduced as an optional coding mode.

4.2 A Brief Introduction to RTP

In this section, we briefly outline the transport protocol used to carry real-time streams on the Internet: the Real-time Transport Protocol (RTP) and its associated Real-time Transport Control Protocol (RTCP) [11] which is in the process of moving from Proposed Standard to Draft Standard in the IETF.

RTP and RTCP are used on top of UDP/IP. The basic idea of RTP is to put a small amount of multimedia data — for example parts of a coded frame or some 100 milliseconds of coded audio — along with control information including an identification of the data type contained in the packet, a sequence numbers and a timestamp into a datagram and transmit it to the receiver. Neither RTP nor UDP mechanisms perform error correction functions and hence a packet will be transmitted at best-effort, i.e. the packet may be received once, but may also be duplicated or lost. The UDP checksum mechanism ensures that all packets delivered to the recipient do not contain bit errors (by discarding all packets that do). The receiver may use the time stamp together with other timing information obtained from RTP and RTCP to dynamically adjust the playout point of the data contained in that packet so that the media stream can be played continuously (with the timing between information units identical to the sender's recording).

This algorithm is only of limited applicability though. As outlined before, applications may have a threshold on the maximum acceptable transmission latency so that a recipient can use the packet contents only in if the timestamp indicates that the received packet is not older than this application specific limit (to ensure timely delivery network layer mechanisms would be required). Packets that arrive too late at the receiver are then treated as packets that were lost in the network. Consequently, the packet loss rate of an RTP-based media stream as observed by the application may be higher than the loss rate caused by packet drops in the network and bit errors.

The minimum header size of an RTP packet is 40 bytes (20 bytes IP, 8 bytes UDP, and 12 bytes basic RTP header). Ignoring the H.263+ payload specific header (refer to section 4.5) for now, from the compression point of view, this yields 40 bytes overhead per packet. For an audiovisual information stream with continuous audio and a video frame rate of 30 frames per second (fps) QCIF with exactly one frame contained in one packet, 30 video packets per second have to be sent to achieve reasonable packetization delay and to avoid queuing in the sending terminal. This results in 1200 bytes per second or around 9.6 kbit/s for packetization overhead (note that a similar overhead is incurred for interleaved audio that is also encapsulated in RTP/UDP/IP).

It should be noted that for the specific case of low speed links, header compression techniques have been developed in the IETF that allow the 40 bytes to be compressed to as little as two or four bytes in the optimal case [12]. However, this compression scheme is only applicable to serial links. Also, this work is not yet published by the IETF as an official standards track document and — although implementation are being done — large-scale deployment of this scheme is likely to take a while. As a consequence, there is a significant interim period for which the aforementioned considerations on overhead are unavoidable.

The primary function of the Real-time Transport Control Protocol (RTCP) for the transmission of real-time media streams is to provide the feedback on the currently perceived transmission quality between a sender and each receiver. This includes dynamic measurement of jitter, throughput, and packet loss rate as well as the estimation of transmission latency and round-trip times. In addition, RTCP allows to estimate the number of participants in a session and provides a means to associate sources of different media streams with a single participant. Finally, RTCP is able to convey a minimum of session control information for informal use (such as names of participants) [13] but also for media stream control (such as requests to transmit an I frame in H.261) [14].

4.3 Control Protocols for Using H.263+

H.263+ has been defined by the ITU-T and its use is defined for virtually all ITU-T conferencing protocol suites including H.323 for IP-based multimedia communications. H.263+ may also be used in conjunction with traditional IETF conferencing protocols and with the IETF multimedia streaming (retrieval) protocol. This section provides some background on the respective system aspects as a basis for later inclusion of the error resilience modes.

In order to make use of H.263+ for video information stream, the applications on the involved endpoints have to determine whether all of them speak H.263+ at all and, if so, which optional modes of H.263+ and — in the context of this paper — which additional error resilience modes they commonly *support*. Based on this outcome, the involved applications have to decide — initially when starting the session and possibly repeatedly during the session to adapt to varying network conditions — on those options they want to *use*. Depending on the control protocols used, these goals are achieved differently — and to a larger or lesser degree.

In the ITU-T context, the H.323 series of Recommendations defines setup and control protocols for multimedia point-to-point calls and small group conferences. A dedicated protocol — H.245 [15] — provides means for negotiating commonly available the (video) codecs, their parameters and optional modes as well as support for further media related algorithms, mechanisms, distribution modes, etc. (including support for error resilience modes for audio and video). This is termed capability exchange. H.245 also provides the mechanisms to initially choose and subsequently change a parameter set for a media using the same language as the capability exchange does. H.323 uses RTP as real-time transport: H.245 describes in advance the information that can and will be carried in RTP, but the RTP / RTCP packets are used to convey information that need to be synchronized with the flow of real-time media (e.g. which type of data is actually contained in an RTP packet).

In the IETF context, a multimedia teleconference as well as a broadcast session is traditionally loosely coupled which means (among other things) that there is nothing like the aforementioned capability exchange and that large groups of participants can be supported. A conference along with its attributes is defined solely by its creator using a session description which is defined by the Session Description Protocol (SDP) [16]. A session description contains fields describing the originator, the purpose, and the lifetime of a session as well as all media to be used in the session along with some of their respective attributes such as encoding to be used, options, and further parameters. Such session descriptions are either conveyed to a (well-defined) audience or are publicly announced. In case of a public announcement — e.g. via the Session Announcement Protocol (SAP) [17] — the session description remains fixed throughout the conference, and there is no well-defined interaction between the creator and the other participants to determine which capabilities the others have. Videophone-call style conversations are set up using the Session Invitation Protocol (SIP) [18], which has a very limited support for capability negotiation. In either case, changing the encoding and/or their parameters relies entirely on the information conveyed in-band in the RTP packets.

Video-on-demand style retrieval of multimedia information is done by means of the Real Time Streaming Protocol (RTSP) [19] which has recently been completed at the working group level of the IETF. RTSP is used by client applications to remotely control media servers and instruct them to play or record specified information streams. The services a media server may offer through RTSP are roughly comparable to the functionality of a VCR. RTSP may use SDP session descriptions to convey media specific parameters between client and server and supports a capability exchange between client and server as well as initial setting and dynamic reconfiguration of the codec used for a media stream and the coding parameters.

5 H.263+ Support for Packet Loss Resilience

As already mentioned above, H.263+ generally uses inter picture prediction for reducing temporal redundancy. This leads to the necessity of always having a correctly decoded reference frame available to allow the successful decoding of the incoming data. The decoded frame will become the next reference frame.

Additionally, H.263+ also uses several prediction mechanisms within one coded frame; e. g. a motion vector for one macroblock is predicted out of the motion vectors of the macroblocks above and left of it, and only the delta between this predicted motion vector and the one to be transmitted is later on Huffman coded. Also, H.263+ uses non-reversible variable length codes, which offer a very high compression ratio, but do not allow resynchronization within the coded data stream, if stream is corrupted. For those reasons, a single picture may be constructed from several segments, which do allow resynchronization of the decoder at their start in the coded data. These segments are either Slices or GOBs (with non-empty GOB headers, see H.263+ for details).

The various optional error resilience modes of H.263+ address both of the above potential synchronization problems in very different manners. They can be considered only as a "toolkit" and have to be used in specific combinations and require transport and control protocol support to be useful. In the following, the modes themselves are described, and later a couple of useful mode combinations are discussed, which offer error resilience for the application types described in section 2.

5.1 Transport oriented modes of H.263+

H.263+ defines five optional modes of operation that deal with transmission errors at the video coding level and are hence referred to as *transport-oriented modes* throughout the remainder of this paper:

- *Forward Error Correction* mode (Annex H) is the oldest of the transport oriented optional modes of H.263. It was already present in H.261 as a mandatory feature; in version 1 of H.263 it was made optional because of its limited usefulness on PSTN/H.324 terminals. If Annex H is used, the complete H.263 bit stream is divided into 'packets' of 492 bits each. A 19 bit BCH forward error correction (FEC) checksum is calculated on all the bits of such a 'packet', along with one bit that allows the resynchronization on the packet structure. This FEC coding allows the correction of single bit errors in each packet and the detection of two-bit errors at a penalty of about 4% increase in the bit rate. However, it is necessary that all bits are transmitted because even one missing bit will make a complete resynchronization to the packet structure necessary, and this may take about 30,000 bits of data (roughly one quarter of a second at a speed of 128 kbit/s). The FEC mechanism of Annex H was designed with ISDN as an isochronous network in mind that also provides a very low error probability. Annex H is not useful in networking environments in which packet losses rather than single bit errors occur (such as the Internet) and is therefore left out in the further discussion.

- *Slice Structured Mode* (Annex K) allows the replacement of the original GOB-layering scheme by the more versatile slices. Slices always consist of a number of macroblocks belonging to the same picture, and every macroblock of a picture has to be assigned to exactly one slice. However, these macroblocks may be arranged either in scanning order, similar to the slice concept of MPEG 1 [20], or in a rectangular shape.

  Scanning-order slices are intended to achieve a – more or less – constant packet size. An encoder can 'fill' the slice with coded data bits until the maximum size is reached, and continue at the next macroblock boundary with a new slice. As any picture segments, scanning-order slices are self contained in such a way, that they are decodable given that a copy of the picture header of that frame is present.

  Rectangular slices are especially useful in conjunction with the Independent Segment Decoding mode, described below. The whole picture can be broken up into as many rectangles as necessary, to achieve reasonable coded slice sizes.

  Using any form of slices has no significant bit rate penalty; the additional bit rates used e. g. for the size information of rectangular slices is well below 1% for virtually all useful cases.

- *Independent Segment Decoding* mode (Annex R), allows the treatment of segment boundaries as picture boundaries. A segment is defined as either a Slice, a GOB, or a number of consecutive GOBs with empty GOB headers. This mode allows the absolutely independent coding of picture parts with no data dependencies between the parts whatsoever if and only if the shape of the independently decodable segments remains identical between two I-frames (as defined in Annex R

of H.263+). This allows the prevention of any error propagation between the various picture segments.

Since Annex R specifies that the shape of any independently decodable segments has to be the same between two I-frames, the use of non-rectangular slices (which are intended to optimize the packet filling) is not very useful. Rectangular slices, however, are very useful in conjunction with Annex R, because they allow to form picture segments with a similar aspect ratio as the whole picture, leading to efficient motion vector coding. This may not be the case if GOBs are used, because Annex R disallows the reference of any data outside the picture segment and thus disallows motion vectors to point outside of the picture segment. To use Annex R on GOBs with non-empty GOB headers leads for all smaller picture sizes (up to CIF) to horizontal-only motion vectors, which hurts the coding efficiency significantly.

The bit rate penalty for using independent decodable segments is highly dependent on the scenery and frame-rate, but in any case substantial. Our simulations show an increase in bit rate of 5% (in case of the sequence Paris) to 15% (in case of the sequence Coastguard), if the CIF picture is coded in four independently decodable rectangular slices of QCIF size.

- *Reference Picture Selection* mode (Annex N), allows to define an arbitrary picture *k* transmitted before (rather than only the latest one *n-1*) to serve as reference picture with respect to which inter picture prediction coding is performed for the new picture *n* to be coded. This mode can be used with or without a back channel. A back channel is a low speed communication channel between from the decoder to the encoder.

  If the Reference Picture Selection mode is used with a back channel, this back channel contains either positive or negative, or both positive and negative acknowledgment information about the successful decoding of pictures including the temporal reference as a picture ID. This informs the encoder about the latest reference picture available at the decoder so that the encoder can decide to use an older picture as the reference picture or to send an I-frame. The back channel can be either multiplexed into the H.263 data stream of the opposite direction or can be conveyed out-of-band. In case of point-to-point connections with low transmission delay characteristics, such as in mobile communications, back channel mechanisms are a valuable addition to achieve temporal error resilience.

  In application scenarios that involve some ten, hundred, or even more endpoints and that are based upon multicast or broadcast communication mechanisms — as available in LANs, intranets, and the (Mbone of the) Internet — back channels are generally not applicable. Back channels are also not acceptable in case of high network latencies.

  When using Annex N without a back channel in a way known as Video Redundancy Coding, temporal error resilience can be achieved, which also can be used in conjunction with the spatial error resilience mechanisms of Annex R and Annex K. See [21] and the discussion of scenario 6 below for a short introduction to Video Redundancy Coding.

  It is possible to apply the Reference Picture Selection mode to single picture segments instead of full pictures. This combination is especially useful in case of large picture sizes (CIF and larger) and high packet loss rates.

  Using older reference pictures than the last transmitted one leads to additional bit rate, because of the higher amount of delta information and the longer motion vectors. In case of an available back channel it is possible to identify situations in which an older reference picture has to be used and thus the reference of older pictures will only be done if necessary, minimizing the bit rate penalty. In such a case, the reference picture selection mode, occasionally augmented by independently decodable segments for larger picture sizes, is the most effective way of handling packet loss situations.

  If, however, no back channel is available and the schemes like VRC have to be used, the bit rate penalty is substantial. VRC needs between 15% and 50% additional bit rate, depending on the scenery and VRC scheme. As a general rule, 40% additional bit rate will result in acceptable quality in 20% packet loss situations (See [22] for details).

- *Temporal, Spatial and SNR Scalability Mode* (Annex O), is the H.263+ specific description of the general concept of a layered codec. One base layer can be enhanced in temporal or spatial resolution as well as in its quality, thus improving the Signal to Noise Ratio (SNR). A temporal enhancement layer consists of B frames known from the MPEG family of video coding standards, whereas spatial and SNR scalability is implemented by coding the differences between the original frame and the decoded base layer in the enhancement layer. Several enhancement layers can be 'stacked' together to form a multi-layered data stream. B-frames, however, are not allowed to be used as anchor points for other information and thus the temporal enhancement always has to be the hierarchically highest enhancement layer. Each layer may be either transmitted individually, or new syntactical elements of H.263+ can be used to multiplex various layers into one data stream. See Annex O for details.

  The Scalability mode is mentioned here, because it provides error resilience in such a way that loss of data inside one layer other than the base layer will only have negative impact on that and all hierarchically higher layers, but not on the basing layers. This is especially useful if more than one virtual connection is available, and the different "connections" have different error characteristics. However, using the layered codec often has negative impact on the coding efficiency, and makes also substantial higher CPU usage than the coding/decoding of a single bit stream with a similar bit rate. In the rest of this paper, that mode will no more be mentioned; however, further work is required to fully realize the potentials of this mode with respect to error resilience support.

5.2 H.263+ Payload Specification

For H.263+ video streams, additional information needs to be carried in the RTP packets containing the respective media information and mechanisms for determining the current network conditions are required. In the IETF, such a payload format specific to H.263+ is being defined [23] which is presented in the following. Two pieces are required for the RTP payload format specification for H.263+: a definition of rules where the continuous bit stream generated by an H.263+ encoder may/shall be divided for packetization on one hand and how those chunks of coded video are placed in packets and which accompanying information is provided in each packet on the other. In the H.263+ payload format, both parts are designed to maximize error resilience against packet loss and to minimize the overhead (in terms of additional header bytes).

For the packetization process, the objectives are to avoid fragmentation at the IP layer as this reduces the packet loss probability and to keep the video packets independent from one another in order to increase robustness in case individual packets are lost. Avoiding IP fragmentation means to keep the size of a video packet (including headers) beneath the maximum transmission unit (MTU) of the internetwork the packets traverse — in general, the size of an Ethernet frame is used as reference yielding an MTU of 1,500 bytes. Keeping video packets independently decodable means to perform semantic fragmentation based on knowledge of the structure of the video stream and optionally adding context information to individual packets where necessary.

The H.263+ payload format specification supports division of the coded video stream at a picture or a picture segment boundary — according to H.263+ this is either a picture, GOB, or slice boundary (identified through the respective start code in the coded bit stream). If an entire frame fits into a packet (the size of which is constrained by the MTU), the entire frame is sent in a single packet. If not, GOB or slice boundaries are sought in the video stream as packetization boundaries. All these packets are called *Picture Segment* packets. If neither a single GOB nor a single slice fits into one packet, pure semantic packetization is no longer possible. In this case, a *Picture Segment* packet is generated and filled up to the MTU limit; the remaining video data is carried in one or more *Follow-on* packets yielding a *packet sequence* of two or more packets.

Note that although *Follow-on* packets are not independently decodable, performing the fragmentation at the RTP rather than the IP layer is advantageous in spite of the UDP/RTP header overhead per packet: the application is in control of the timing for sending out the packets can thus is able to avoid bursts that may lead to increased packet loss; Furthermore, if one packet out of such a sequence is lost the remaining ones are still received by the receiver and may — depending on their contents — at least

partially be processed; and, finally, the UDP header is required to identify packets as belonging to a packet flow if RSVP resource reservation mechanisms are used.

The H.263+ payload specification also defines an additional header that immediately follows the standard RTP header. Together with the first six bits of the coded video payload, the H.263+ specific header identifies the packet type — Picture Segment (and segment type) vs. Follow-on packet — and allows the inclusion of additional information:

- If Video Redundancy Coding (VRC) is used, a VRC extension header immediately follows the H.263+ header that contains information which enables the receiver to detect damaged threads at the transport layer (i.e. without the need to decode the video stream). This allows to feed only undamaged thread data to the decoder and thus avoid the additional CPU load, which would be necessary in other cases. See [21] for a detailed description of VRC.
- To any Picture Segment packet, a copy of the current picture header may be added to provide additional information that may be required for independent decoding — e.g. in case the previously sent picture header was lost. This mechanism can also be used to repeatedly send the full picture header instead of the abbreviated picture header (UFEP mechanism), which may be useful for gateways. In a pure IP environment, however, no abbreviated picture headers should be used.

Figure 1 depicts the payload of a UDP packet containing H.263+ encoded video with all optional information fields present: the P=1 indicates that a picture or picture segment start is contained in the packet (in this case, the first 16 zero bits of the respective start code are not included in the video bit stream but have to be prepended by the recipient). The V=1 indicates that VRC information is present as well. The VRC information field contains the thread id (TID) and the packet sequence number (TRUN) per thread and an indicator if the frame is a synchronization frame (S=1). If PLEN>0, PLEN bytes following the (extended) H.263+ payload specific header carry the picture header; in this case, the PEBIT field indicates how many bits of the last byte belonging to the picture header are unused.

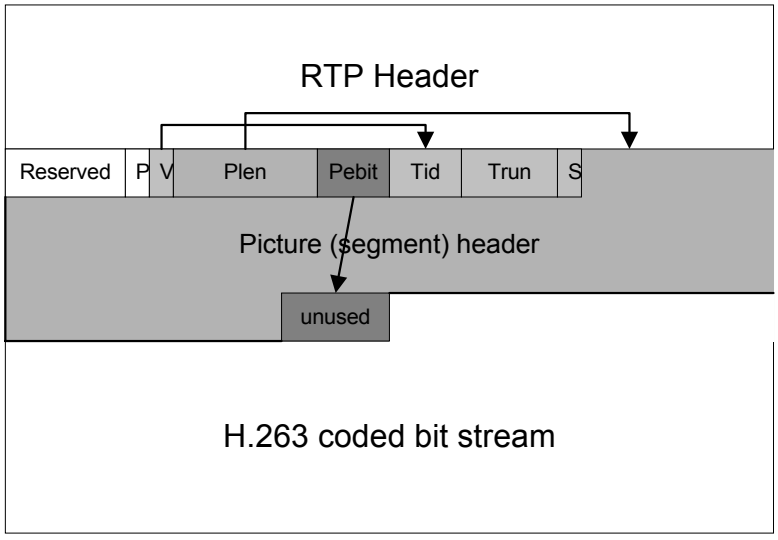

Figure 1: RTP payload format for H.263+

If the Scalability mode (Annex O) is in use, the packetization is used on the pictures of each layer independently, thus leading to as many data streams as layers. Mechanisms outside the scope of the payload spec have to be used to define the dependencies of the various layers. It is also allowed, to place more than one layer into one packetized data stream by using the layer multiplexing mechanisms of H.263+, but that is neither very useful nor recommended by the payload specification.

6 Protocol Support for Packet Loss Resilience

The requirements to support error resilience against packet loss in the wide area Internet are at least threefold: i) first of all, appropriate feedback needs to be gathered from which the current network conditions can be derived; ii) means for taking corrective action — as outlined in the previous section — need to be supported by transport and control protocols; and iii) video applications themselves have to actually make use of the measurements and the mechanisms provided by the protocol infrastructure. The first two aspects are discussed in this section. A feasible set of rules for application behavior is presented in section seven.

6.1 Feedback Mechanisms

Various generic monitoring functions are already supported by RTCP as briefly introduced in section 4.2: one-way delay and round-trip time can be estimated by senders and receivers; inter-arrival jitter, absolute and relative packet loss, and achieved throughput are measured per sender at each receiver and are reported back to the senders. This is achieved through the RTP senders transmitting sender and RTP receivers transmitting receiver reports in regular intervals. In order to allow scaling to large groups, the bandwidth used for RTCP is limited to a fraction of 5% of the overall bit rate "allocated" to the RTP session. As all members in a session, observe the RTCP messages and thus learn about the number of session members, they are able to adapt their transmission rate for RTCP reports accordingly.[4] The RTCP mechanisms are independent of the media stream encoding and provide a reasonable estimate of the current network conditions which is fed back from the receivers to the senders on a regular basis; note that the latter implies that receivers are generally not able to provide immediate feedback to a sender via RTCP.

In addition to the media-independent feedback, the Reference Picture Selection mode of H.263+ optionally envisions a back channel to report parts of the reception and decoder status back to a sender. For example, the receiver may notify the sender about the loss of a particular frame so that the sender can use a different frame for prediction information for encoding the next frame to avoid error propagation.

H.263+ provides mechanisms to carry the back channel within the encoded video (of the opposite direction) but out-of-band mechanisms may be required as well: if the video transmission is one-way as in retrieval or broadcast applications; if feedback shall be given to multiple senders (via multicast) as in conferencing applications; or just because the desire is in a modular system design to decouple (transport level) control from video coding functions. The ITU-T Recommendation H.245 defines a dedicated message — which is delivered reliably but may be delayed because of retransmissions, interleaving with other messages, and possible indirect routing of H.245 messages — to carry H.263+ back channel information and thus provides the out-of-band mechanism for all H.323 terminals. In the absence of H.245 — as is the case for loosely-coupled conferences, broadcasts, and RTSP-based retrieval applications — or if more timely feedback is desired, other mechanisms have to be employed.

RTCP may be extended by a payload format specification to also carry media specific information. An example is the RTP payload for H.261 coded video streams: two additional control packets are defined to request an intra frame from the sender (Full Intra frame Request, FIR) and to notify the sender that particular packet was not received (Negative Acknowledgement, NACK) [14]. A similar approach could in principle be used to carry the back channel of H.263+ as well. However, as RTCP packets may only be sent at a certain maximum rate a receiver may not be allowed to provide immediate feedback to the sender — and the benefit that can be derived from the feedback decreases sharply over time. Hence, using RTCP in its current form is only of limited value.

An alternative to be considered is the extension of the H.263+ payload itself to carry the back channel information. The authors have originally proposed to piggyback the back channel in RTP media packets in case of bi- or multi-directional video communications as this would allow timely feedback

[4] Additional mechanisms such as random dithering the reports [RFC1889] and timer reconsideration [30] are employed to prevent bursts of reports from occurring because of synchronization effects during normal operation but particularly at the beginning or end of a session (or whenever the group size changes significantly in short period of time).

at almost no overhead. However, in case of unidirectional communication, a different mechanism — embedding in RTCP or RTSP — would still be required.

At the time of writing, there is still an on-going debate in the Audio-Visual Transport Working Group of the IETF on how to implement the back channel in the context of RTP. In particular, provision of a generic feedback mechanism including appropriate changes to RTCP if required is envisaged that is not restricted to be used with H.263+. For the specific case H.263+, the impact of increased feedback delay on coding efficiency and picture quality of subsequent frames needs to be investigated further. Also, not providing a H.263+ back channel at all (for certain applications) has to be taken into consideration.

Based on our current knowledge, in our opinion, the following use of feedback mechanisms for the application types is appropriate. The generic RTCP mechanisms are applicable to and should be used by all application types to derive which error resilience mechanisms to use. Then, media and error resilience specific feedback information may be used in addition, to fine tune the transmitted video stream.

For point-to-point video telephony (a1), network scalability is not an issue and neither need be the RTCP delay. Assuming a bandwidth of 80 kbit/s, RTCP may consume a total of 4 kbit/s per second, i.e. 2 kbit/s per participant. An RTCP sender report in a session with two parties is 80 bytes in size; assuming another 10 bytes to carry the back channel information, this yields 90 bytes or 720 bits per message. Hence each peer may send three back channel messages per second with enough flexibility in scheduling so that virtually immediate feedback can be achieved. Furthermore, assuming a worst case frame rate of 30 fps with one frame per packet (QCIF image size) at a worst case of 20% packet loss,[5] in the average no more than six packets should be lost per second so that losses can be reported back in a timely fashion. In any case, piggybacking the back channel messages on the RTP packets is a perfect alternative in this scenario.

For multipoint video conferencing (a2), even in small groups, the value of the back channel is disputable. Researches have found that when using the current Mbone for multicast distribution, packet losses observed at different receivers are typically not correlated — unless the packet was dropped close to the source [6] [9]. Uncorrelated loss, however, means that the sender may not be able to make a decision that is beneficial to the entire group of receivers. Also note that the sender may want to accumulate feedback from several endpoints and in this case has to decide how long to wait for messages to arrive before taking action based upon the feedback. To minimize the delay and maximize the available information, it seems reasonable to follow the piggybacking approach rather than to use RTCP — if a back channel shall be used at all.

In retrieval applications (b) control is typically exercised by a single receiving endpoint but the media stream may be multicast (e.g. into a conference). Depending on the number of recipients, the suggestions for either a1 or a2 apply — with the exception that no piggybacking of feedback is possible and hence RTCP would have to be used. For the endpoint controlling the media server, the RTSP channel may be possible feedback mechanism as well.

In broadcast applications (c), support for a back channel does not seem to be useful at all. The only mechanism available is RTCP but the transmission interval for receiver reports are potentially large and hence the sender would base corrective actions always on the feedback from a subgroup of receivers which may not be representative.

6.2 Adaptation Signaling Mechanisms

Whenever a sender of a video stream decides to adapt the encoding and to enable or disable (not only) error resilience mechanisms, this impacts the recipients' processing of the media packets. The H.263+ video bit stream is always self-contained and hence, in principle, no additional notification mechanisms are needed — unless the nature of the transmitted bit streams changes beyond what is visible to H.263+ and / or the respective information shall be made visible outside the video coding module (e.g. at the transport layer) as well. This applies to the following error resilience modes:

[5] As stated before, a packet loss probability of some 20% is roughly the upper bound where known error resilience modes for interactive video communications still work (refer to section 5.1 and to [22]).

- *Layered coding*: if the sender decides to turn layered coding on or off, or to change the number of layers or their respective meaning, this needs to be signaled out of band. In particular, if layers are added or their transport addresses are changed, the new / changed addresses have to be conveyed to the recipients.
- *Video redundancy coding*: if VRC is turned on or off, or if the number of threads changes, this should be signaled out-of-band to the receivers. The receivers may as well determine presence or absence of video redundancy encoding from the H.263+ bit stream, but finding out the number of different threads requires one packet of each thread to be received.

For conferencing systems based on H.323 (only covering application types a1 and a2), the necessary signaling facilities are provided by H.245. For loosely coupled conferences (a2), retrieval applications (b) and broadcast applications (c), all of which are solely based on IETF protocols, these facilities are not available. This has the following implications for the use of layered coding: loosely coupled conferences (a2) and broadcast sessions (c) must initially specify the maximum number of layers and provide all the necessary transport and semantic information about each layer; also they are not allowed to change these parameters dynamically (an application may decide to reduce the number of layers temporarily by not sending any further data to a particular address). For small group conferences, an additional control protocol — such as the Simple Conference Control Protocol (SCCP) [24] — may be used to dynamically (re-)configure media streams. Retrieval applications (b) may make use of RTSP mechanisms to change the media stream (session) description dynamically and hence have in principle the same flexibility as an H.245 based application.

For Video Redundancy Coding, SCCP or RTSP may be used instead of H.245 to carry configuration information — but VRC works without any of those protocols as well. All application types may make use of in-band signaling in the RTP header as provided for Video Redundancy Coding. The VRC information contained in the H.263+ payload header allows determining the number of threads from looking at the thread ids in the payload header without having to consult the decoder module.

7 Considerations on the Application of Error Resilience Modes for the Application Scenarios

Wenger et al. [22] derived preferred mode combinations for interactive applications in various network scenarios. This section recaps these findings and augments them in respect to the wider field of application types for the wide area Internet as the primary networking scenario and also considers additional findings on error resilience described by Wenger et al. [25].

Out of the thoughts above, three basic decision processes regarding the mode combinations of H.263+ for a given application and network situation can be identified. These three decision processes are:

I) Based on the *application type* and the latency requirements of the application the suitability of using back channel mechanisms has to be investigated. Back channels do, if advisable, offer the best available error resilience and compression ratio, but are only feasible for certain interactive and rare semi-interactive applications, as described in some detail in section 6. This decision can be made early, often as early as the design and compile time of that specific application.

II) The selection of the *picture size* is often a function of the user interface, and will change only infrequently during an established connection. Similar assumptions are valid for the *target bit rate* for the coded picture and for the target *frame rate*. These three parameters can be used to make decisions on the number of the independently decodable segments the picture should be divided into.

III) During the duration of a communication relationship, the experienced packet loss may change significantly without the user's influence; even if traffic shaping techniques — e.g. avoiding bursts — are used to attempt keeping the packet loss probability to a minimum. Given a certain packet loss rate and also information from the coder on the amount of motion in the coded scenery, the coder can adjust the I-frame/I-segment rate and decide on the use and parameterization of VRC.

The following subsections describe algorithmic rules that seem to be appropriate for each decision.

7.1 Decision Process I: Application Type

The criteria for using a back channel based on the application type was already discussed in section 6 in some detail as were the conclusions. In general, for interactive and some semi-interactive applications, back channel mechanisms are appropriate and should be used, provided, that the codecs implement those mechanisms. For most semi-interactive, video-on-demand style applications with pre-recorded data, as well as for broadcasts, back channels are not appropriate — for the reasons discussed in section 6.

7.2 Decision Process II: Picture Size, Target Frame Rate, and Target Bit Rate

Assuming a constant advisable MTU size (e. g. 1500 bytes in case of the Internet), it is easy to calculate the typical number of packets per frame at a given bit-rate and frame-rate, including some percentage for variances. For this calculation, I-frames can be left out. If such a calculation shows, that typically more than one packet is needed for one coded frame, then the picture should be divided into as many independently decodable segments as needed, so that each segment fits into one packet. Moreover, the packetization should be used appropriately to add the redundant picture header to every packet, thus allowing the independent decoding of picture parts, even if earlier transmitted parts of the picture got lost. Other error resilience mechanisms (VRC or back channel), as used following decision process III, should be applied on those independently decodable segments rather than on entire frames. Only in the rare case of communications at a guaranteed quality of service with no packet loss at all, the bit rate penalty of using independently decodable segments can be avoided.

7.3 Decision Process III: Observed Packet Loss

The encoder can be kept informed by the transport hierarchy about the packet loss situation in the network (e.g. by the means of RTCP receiver reports). Additional information about decoder states (like Full Intra frame Requests (FIR) or back channel messages) may also be available in specific applications. From this information, the encoder has to choose an appropriate coding mode. In the following, a set of rules on how to deal with different packet loss rates is described: first, special cases for the optimization of I-frame transmission and back channel operation are discussed, afterwards the general case (in absence of I-frames and back channel) is presented.:

Special case I-frame: If an I-frame is to be sent such an I-frame typically is between 7 and 12 times larger than the usual P-frame and generally does not fit into a single packet. Therefore, I-frames should be divided into packets by using either scan-order slices (preferred) or the GOB-mechanism (if no slices are available). The packetization should always add a redundant picture header so that as many packet as possible can be independently decoded, even if a part of the I-frame is lost during transmission. Signaling Annex R in case of I-frames is neither necessary nor has any positive or negative effect. If decision process II) has decided on using several independently decodable segments, those segments should not be sent altogether as one I-frame, but interleaved as more than one I-segment. This rule does not apply if the I-frame is transmitted in response to a FIR.

Special case Back Channel: If back channel mechanisms are available the error resilience functionality should completely rely on the back channel mechanisms. The I-frame/I-segment frequency should be chosen high enough, to give synchronization points in such cases in which back channel data got lost (which itself obviously depends on the reliability of the back channel itself and thus cannot be defined here, because the back channel may be transmitted over protocols with different characteristics). If the picture was divided into segments according to the decision process II, then the back channel mechanisms should be applied to each segment individually. Note, however, that the back channel is no longer applicable if the typical round trip type for forward data and back channel message becomes larger than the mean packet loss interval. In such a case, the mechanisms as described below are in charge.

General case: The following rules apply in absence of a back channel and for all frames except I-frames.

1. If the observed packet loss rate is below a threshold of 3% to 5%, an I-frame interval of half the mean packet loss interval should be used for good quality needs and an I-frame interval equal to the mean packet loss interval for usual quality needs. If decision process II) chooses to break up the picture into more than one segment, a similar mechanism should be applied for each segment. A complete I-frame should never be sent; instead, interleaved I-segments should be transmitted — except a FIR was received.

   The threshold of 3% to 5% depends on the amount of motion (which can be easily determined from observing the Intra-block rate and the typical number of motion vectors and their respective lengths). When more motion is observed, the threshold should be lowered and vice versa. For test sequences like Paris, 5% is well applicable, whereas for the test sequences Foreman, Coastguard, and Stephan, 3% are more useful.

2. If the observed packet loss rate is higher than the one given in (1.) and lower than 20%, either the I-frame/I-segment rate may be raised similar to III.1) or Video Redundancy Coding may be applied. VRC will only show a marginally better compression ratio, but much better real-time characteristics and also better subjective quality due to the more frequent use of motion vectors, that reduce number of blocking artifacts. VRC itself may be scaled as follows (which showed acceptable results for 'Talking head' sequences like Paris): for loss rates lower than 7% use 2 threads at 5 pictures per thread, between 7% and 15% use 2 threads at 3 frames per thread, for higher rates use 3 threads at 3 frames per thread.

3. If the packet loss rate exceeds 20%, the mechanism defined in III.1) should be used again. In general, only I-frames should be used in such a case: At such a packet loss rate, the reference frame will often not be available. In addition, the size difference between I and P-frames may result in negative visual effects: it takes significantly longer to receive and decode a complete I-frame compared to a P-frame before the receiver is able to display it. I-frames and P-frames would be displayed for different durations leading to alternating update bursts and "still images" — unless significant extra playout delay is introduced at the recipients to compensate for this.

8 Simulations / Test Results

Two types of simulations and tests were performed to verify the contents of this paper:

- tests on several long distance and intercontinental connections on the Internet showed packet loss characteristics for various packet sizes, bit-rates and
- simulations using a modified public available H.263+ codec delivered data on the effectiveness of the mode decision process, as described in section 7.

At the first glance, the two tests seem to have little correlation. However, the video simulations were done based on the knowledge about optimal MTU sizes, transmission delays, bit-rates and packet-loss characteristics as found during the network tests. In this paper, we can only give a very brief overview on all the various tests we performed; however, both test software and technical reports describing the results are available [by the time this paper is published] from ftp://kbs.cs.tu-berlin.de/local/kbs and ftp://ftp.tzi.uni-bremen.de/tzi/dmn/projects.

8.1 Packet loss on the Internet

We conducted three test sessions each for a roughly 24 hour period. The sites involved were the TU Berlin on one side acting as sender and the University of Bremen, the University College London, and the MIT on the other acting as reflectors for the packets. Route traces using the *traceroute* command showed distances of 9, 15, and 22 hops between the sites, respectively. All tests consisted of the transmission of UDP datagrams from Berlin to the respective receiving site which reflected the datagram back and collected reception statistics. Sequence number and time stamp information contained in the packets allowed to measure round-trip time, packet loss probability, and packet loss distribution. The sending program emitted datagrams at regular intervals (each 100ms) at payload

sizes ranging from 128 to 1,468 bytes for individual packets. Payload sizes of 2,048 and 4,096 bytes were used to simulate two and three packet sequences, respectively, because of IP layer fragmentation. Through the variation of the packet size bit rates ranging from 8 kbit/s to some 120 kbit/s were simulated for packet sequences of length one. Each test run consisted of 100 packets and was carry out approximately once per hour for each packet size.

For one or two packet sequences, the maximum packet loss in a sequence of 100 packets was 22 % for Bremen with an average of around 2.5 %, 28 % for UCL with an average of 4.5 %, and some 30% for the MIT with an average of around 10%. For transmission rates between 8 kbit/s and 120 kbit/s, no correlation between the packet loss probability and the transmission rate could be found. For three packet sequences, the minimum datagram loss rate was 40%, the maximum between 53% and 75% at an average of around 50% roughly indicating a packet loss rate of more than 20 %.

As already outlined in section 3.2, similar findings have been reported in other studies as well. It should be repeated that these findings roughly apply to multicast as well and that additional investigations for multicasting have shown that packet loss between receivers is largely uncorrelated, too.

8.2 Video simulations

In section 6, three decision processes were outlined and justified theoretically. On some of the probably not yet well understood aspects of those decision processes we performed non-real-time simulations to verify our thoughts in some way, however, more work will have to be invested in this area. We invite everybody to download our real-time software (an enhanced version of vic [26] that supports H.263+ and all mechanisms of section 6 except the back channel) and our non-real-time software (H.263+ codec supporting tmn8 and most of the optional modes, based on the reference implementation of UBC [27]) and comment on our results.

Decision process I, which used the application type to decide on the use of back channel mechanisms, was not simulated for this paper. However, various core experiments of the ITU-T experts on low-bit video coding (now SG16 Q.15) showed the superiority of back channel mechanisms, if available over any other error resilience tool [28]. This is especially true, if those mechanisms are augmented by simple error-concealment techniques in case of using more than one picture segment [29].

Decision process II settled the number of independently decodable picture segments of one picture, based on picture size, bit rate, and MTU size considerations. This is not done to improve coding efficiency or spatial picture quality significantly, although positive implications on those factors are also observed. The reason for this segmentation is to allow as small I-segments as possible in order to prevent error propagation across those segment boundaries. Small I-segments lead to a generally lower coding delay, and this reduces the overall latency. The only tests we carried out on decision process II so far was to investigate the bandwidth overhead for using independently decodable picture segments. We found a bit rate penalty of some 5% to 15% (depending on the scenery) if four picture segments are used. This leads to an SNR decrease of probably 2 dB at constant bit rates, again depending on the scenery, and the shape of the segments. The aforementioned technical report provides a more detailed discussion on those issues as did [22].

In this paper, only decision process III shall be discussed in some more detail. While, again, any back channel mechanisms were left out for the reasons given above, the gain of employing Video Redundancy Coding instead of the repeated transmission I-frames is shown in the following.

The following simulation results were generated using a common set of parameters (unless indicated otherwise):

- The modified UBC reference software codec was used for the simulation. The only modification worth mentioning was the inclusion of the Annex N functionality, but with a different syntax.
- The sequences Foreman, Coastguard and Paris all of them in QCIF image size were used. A fixed quantizer of 10 and a fixed frame rate were applied which resulted in a variable bit rate. QCIF image size was used because both the bit rates and the necessary CPU power seemed to be useful for software based codecs and Internet connections

- None of the optional coding modes except our implementation of the functionality of Annex N was employed. The results should be similar to more complex mode configurations, although the bit rate should be somewhat lower. Some information regarding the benefits of using the quality oriented optional modes of H.263+ as well as their impact on the complexity can be found in [27].
- The rate of I-frames was chosen in such a way that an I-frame was sent every 5 seconds, regardless of the frame rate.
- All sequences were looped for the duration of 5 minutes to prevent the influence of the random nature of the packet loss generator. Additional simulations showed that that the worst case influence is still in a 0.4 dB interval, the average somewhere less than 0.2 dB.

Sequence, Frame rate	VRC-Scheme	Packet Loss	Data rate (Kbit/s, %)	SNR	Remarks
Paris	None	None	82.3 ⇒100%	29.6	Error free environment, no VRC
Paris, 10 fps	None 2-2 2-5 3-3	10%	82.3 ⇒ 100% 111.0 ⇒ 135% 104.3 ⇒ 127% 121.8 ⇒ 148%	26.6 28.7 27.8 28.5	 2.2 dB SNR for 27% more bit rate 2.9 dB SNR for 48% more bit rate
Paris, 10 fps	None 2-3 3-3	20%	82.3 ⇒ 100% 111.0 ⇒ 135% 121.8 ⇒ 148%	24.4 26.5 26.8	 2.1 dB SNR for 35% more bit rate 2.4 dB SNR for 48% more bit rate
Paris, 30 fps	None 2-3 2-5 3-3	10%	148.8 ⇒ 100% 214.6 ⇒ 144% 204.2 ⇒ 137% 248.8 ⇒ 167%	25.6 28.4 27.7 28.3	 2.8 dB SNR for 44% more bit rate
Paris, 30 fps	None 2-3 3-3	20%	148.8 ⇒ 100% 214.6 ⇒ 144% 248.8 ⇒ 167%	23.6 26.0 26.2	 2.4 dB SNR for 44% more bit rate
Foreman, 15fps	None 2-2 2-3 3-3	10%	92.8 ⇒ 100% 128.2 ⇒ 138% 125.0 ⇒ 135% 144.9 ⇒ 156%	24.4 27.5 28.1 28.7	 3.7 dB SNR for 35% more bit rate 5.3 dB SNR for 56% more bit rate
Coastguard, 15 fps	None 2-2 2-3 3-3	10%	190.6 ⇒ 100% 252.7 ⇒ 133% 249.2 ⇒ 131% 280.0 ⇒ 147%	22.9 26.8 26.9 27.0	 4.0 dB SNR for 31% more bit rate 4.1 dB SNR for 47% more bit rate

Table 1: VRC simulation results

A second set of simulations was run to check whether or not a similar SNR gain can be reached by simply spending the additional bit rate for a higher amount of I-frames or by using a numerically lower QP. Out of those simulation results, we only present the following examples that are based on a packet loss rate of 10%.

The first three columns of this table do not need any further explanation. The fourth column contains the SNR using a variable number of I-frames to fill the bandwidth overhead introduced by VRC. A modified TMN5 rate control mechanism was used for that simulation. The data was generated assuming one lost packet (with a payload size of 1400 bytes) in an I-frame that leads to the loss of the whole I-frame. While this simulation seem to be somewhat unfair because large parts of the I-frame can still be processed in case of a lost packet, it should be mentioned, that having Intra-updated parts of a picture together with very old non-Intra updated packets in the same picture leads to substantial visual distortions, which are not measurable by SNR mechanisms, but have a great impact on the

visible quality. The technical report also contains data out of subjective quality assessments as well as data generated by a more objective picture quality scale (PQS, [5]).

Sequence and Parameters	SNR for VRC Scheme, 5s I-Frame Interval	SNR, variable QP, 5s I-Frame Interval	SNR, variable number of I-Frames, packet size 1400 bytes
Paris, 10 fps, 111 kbit/s	VRC 2-2: 28.7	27.1	27.0
Paris, 30 fps, 214.6 kbit/s	VRC 2-2: 28.4	26.5	27.4
Foreman, 15 fps, 125 kbit/s	VRC 2-2: 28.1	25.0	28.4
Coastguard, 15 fps, 249.2	VRC 2-3: 26.9	23.2	26.7

Table 2: VRC compared to other mechanisms

Generally, VRC showed in this table a similar performance for low-motion sequences like Paris, and a much better performance in case of high motion sequences like Coastguard or Foreman.

9 Conclusion

This paper has extended earlier considerations on preferred video coding mode combination for various networking scenarios [22] by addressing retrieval and broadcast style video applications as well and by examining transport and control protocol issues in more details — with the exclusive focus on IP-based networks (specifically the wide-area Internet). In particular, the importance of combining the H.263+ video encoding itself, the transport layer mechanisms (namely the H.263+ payload format), and higher layer control protocols to achieve error resilience has been described. A broad range of possible feedback mechanisms to obtain information about the varying network conditions has been introduced including the latest discussions (which are driven by the authors) in the relevant working group of the IETF. The feasibility of these feedback mechanisms have been examined with respect to the three aforementioned video application types.

Together, transport, control, and feedback mechanisms provide the foundation for achieving maximum error resilience for video streams in the Internet. Based on the mechanisms described in this paper, algorithms have been outlined that derive the appropriate combinations of error resilience mechanisms to be applied from the application type and the network conditions. Most of these mechanisms have been implemented in an enhanced version of the video tool vic and fist real-world tests yielded significant improvements of the subjectively perceived video quality if the error resilience mechanisms were applied. To further verify the results, various simulations have been carried out — with this paper focusing on the use of video redundancy coding not only for interactive applications. The parameters used in the simulations were all based upon input from a variety of tests conducted on packet loss characteristics in the Internet.

The simulations carried out so far show encouraging results: the rules derived for applications achieve their goal by providing good guidelines for implementers of packet-based video applications. However, further work is still needed to gather more encompassing statistics on packet loss characteristics — also including non-research parts of the Internet — and to enhance the statistics collection by more detailed information about burstyness of packet loss and transmission delay. Based on this additional knowledge, the rules outlined in this paper may then be extended to cover even more details of the network conditions (such as temporal aspects of variations in the observed transmission quality, lengths of packet loss bursts, etc.). Further simulations are then needed to verify the rules defined in this paper as well as (enhanced) rules for a wider range of scenarios, and further implementation experience is needed to see how far more complex rule sets can be incorporated into network-aware codecs and applications

10 References

[1] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis. "RTP Payload for Redundant Audio Data." Proposed Standard. RFC 2198. September 1997.

[2] J. Rosenberg and H. Schulzrinne. "An AVT Payload Format for Generic Forward Error Correction." Internet Draft draft-ietf-avt-fec-01.txt. Work in progress. November 1997.

[3] C. Perkins and O. Hodson. "Options for Repair of Streaming Media." Internet Draft draft-ietf-avt-info-repair-01.txt. Work in progress. November 1997.

[4] C. Partridge. "A Proposed Flow Specification." Informational RFC 1363. September 1992.

[5] M. Miyahara, K. Kotani, V. R. Akgazi. "Objective Picture Quality Scale (PQS) For Image Coding". submitted to IEEE Transactions on Communication.

[6] M. Handley. "An Examination of Mbone Performance." UCL/ISI Research Report. January 1997.

[7] J.C. Bolot and A. Vega-García. "Control Mechanisms for Packet Audio in the Internet." Proceedings of IEEE Infocom '96. pp. 232-239. San Francisco, CA. April 1996.

[8] J.C. Bolot and A. Vega-García. "The Case for FEC-Based Error Control for Packet Audio in the Internet." To appear in ACM Multimedia Systems.

[9] M. Yajnik, J. Kurose, and D. Towsley. "Packet Loss Correlation in the Mbone Multicast Network." Proceedings of the IEEE Global Internet Conference. London. November 1996.

[10] Generic coding of moving pictures and associated audio information: Video, ISO/IEC International Standard 13818-2, 1995

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. "RTP: A Transport Protocol for Real-time Applications." Proposed Standard. RFC 1889. January 1996.

[12] Stephen Casner and Van Jacobson. "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links." Internet-Draft draft-ietf-avt-crtp-04.txt. Work in progress. November 1997.

[13] H. Schulzrinne. "RTP Profile for Audio and Video Conferences with Minimal Control." Proposed Standard. RFC 1890. January 1996.

[14] T. Turletti and C. Huitema. "RTP Payload Format for H.261 Video Streams." Proposed Standard. RFC 2032. October 1996.

[15] M. Nilsson (ed). "Control Protocol for Multimedia Communication." Draft of ITU-T Recommendation H.245. September 1997.

[16] M. Handley and V. Jacobson. "SDP: Session Description Protocol." Internet-Draft draft-ietf-mmusic-sdp-05.txt. Work in progress. November 1997.

[17] M. Handley. "SAP: Session Announcement Protocol." Internet-Draft draft-ietf-mmusic-sap-00.txt. Work in progress. June 1996.

[18] M. Handley, H. Schulzrinne, and E. Schooler. "SIP: Session Initiation Protocol." Internet-Draft draft-ietf-mmusic-sip-04.txt. Work in progress. November 1997.

[19] H. Schulzrinne, A. Rao, and R. Lanphier. "Real Time Streaming Protocol (RTSP)." Internet-Draft draft-ietf-mmusic-rtsp-06.txt. Work in Progress. November 1997.

[20] Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbit/s, ISO/IEC International Standard 11172, 1992

[21] Video Redundancy Coding in H.263+, Proceedings of AVSPN 97

[22] Error Resilience Support in H.263+, submitted for publication in IEEE Transactions on Circuits and Systems for Video Technology

[23] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, and C. Zhu. "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263Video (H.263+)." Internet Draft draft-ietf-avt-rtp-h263-video-01.txt. Work in progress. January 1998.

[24] C. Bormann, J. Ott, and C. Reichert. "Simple Conference Control Protocol." Internet Draft draft-ietf-mmusic-sccp-00.txt. Work in progress. June 1996.

[25] Two new usage forms of the reference picture selection mode of H.263+ for low latency interactive applications (working title), submitted to DCC'98

[26] S. McCanne and V. Jacobson. "vic: A Flexible Framework for Packet Video." Proceedings of ACM Multimedia '95. Berkeley, CA. November 1995.

[27] G. Cote, B. Erol, M. Gallant, F. Kossentini: "H.263+: Video Coding for Low Bit Rates", submitted for publication in the special 1998 issue of the Transactions of Circuits and Systems for Video Technology

[28] T. Nakai, Y. Tomita: "Core Experiments on Back-Channel Operation for H.263+" ITU-T SG15 contribution LBC 96-308, November 1996

[29] H. Kimata, Y. Tomita, H. Ibaraki, and T. Ichikawa "Concealment of Damaged Are for Mobile Video Communication", Proceedings AVSPN97, Aberdeen, U. K., 1997

[30] J. Rosenberg and H. Schulzrinne. "Timer Reconsideration for Enhanced RTP Scalability." Internet Draft draft-ietf-avt-reconsider-00.txt. Work in progress. July 1997.