Multimedia Systems (2005) 10(5): 402–412
DOI 10.1007/s00530-005-0169-4

RESEARCH ARTICLE

Jari Korhonen · Ye Wang · David Isherwood

Toward bandwidth-efficient and error-robust audio streaming over lossy packet networks

Published online: 7 July 2005
© Springer-Verlag 2005

Abstract Bandwidth efficiency and error robustness are essential issues for different multimedia streaming applications. This paper presents strategies for high-quality audio streaming based on fragmenting perceptually coded audio frames and shuffling the data components among multiple packets for transportation. This is done to increase robustness against packet loss. We also address the delivery of audio data consisting of components with different proportional priorities. Our approach is rationalized with streaming tests using the MPEG AAC audio codec in a simulated network environment and formal listening tests to evaluate the resulting audio output. According to the results, the proposed schemes improve audio quality significantly with reasonable increase to network resource utilization compared to traditional error robustness measures.

Keywords Audio streaming · Multimedia networking · Perceptual audio coding · Teleconferencing · Advanced audio coding (AAC)

1 Introduction

Rapid evolution of IP networking is turning the Internet into a medium for diversified audiovisual content distribution competing with traditional telephone networks and even cable TV. The MP3 file format first started the era of digital music distribution via the Internet some years ago. Nowadays, network connections are still improving and real-time multimedia streaming is gaining popularity among consumers of modern entertainment services.

Different applications set different requirements for quality and interactivity. Voice over IP (VoIP) applications, suited to replacing traditional telephone services, require a high level of interactivity, which sets strict limitations for buffering and transport delay. Therefore, packet loss recovery in VoIP often relies on forward error correction (FEC) rather than retransmission in the network. Multimedia broadcasting and multicasting applications – such as Internet radio or teleconferencing – can tolerate higher transport and buffering delays. Nevertheless, network-based error robustness measures such as retransmissions are still undesirable due to the feedback implosion problem discussed later in this paper. Quality requirements are typically even higher for one-to-one audio- and video-on-demand (AoD, VoD) applications. Luckily, these applications usually also tolerate higher transport and buffering delays, which makes it possible to use different schemes to improve robustness against packet loss, such as retransmissions and interleaving. A rough classification of multimedia content distribution applications according to their requirements for latency and quality is shown in Fig. 1. The schemes proposed in this paper are mainly designed for applications such as AoD, teleconferencing, and Internet radio.

Most existing error robustness techniques, such as the error resilience (ER) tools in the MPEG standards, are designed to combat bit errors. There is also a considerable body of literature on receiver-based error concealments. The aim of receiver-based error concealment is to reproduce

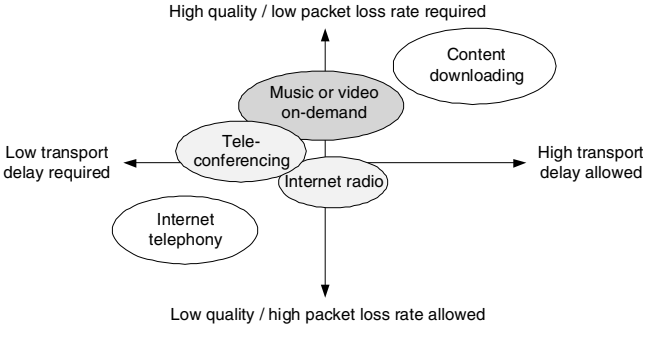


Fig. 1 Different multimedia content delivery application types

J. Korhonen (✉)
Nokia Research Center, Tampere, Finland
E-mail: jari.ta.korhonen@nokia.com

Y. Wang
National University of Singapore, Singapore

D. Isherwood
Nokia Corp., Tampere, Finland

missing audio clips independently, without support from communications protocols [1, 2]. Bit error resilience methods can be greatly beneficial in circuit-switched communications, especially in the wireless domain [3–6]. However, these techniques are not effective against packet loss in packet-switched networks, which is the main concern in this paper. There are no formal methods in the MPEG compression standards specified to combat packet losses.

Many advanced proposals for optimized high-quality audio streaming have been designed for certain carrier networks, such as EGPRS [7] or 3G [8]. Less attention has been paid to robust packetization and transport schemes for streaming audio over generic IP-based packet networks. There are both codec-specific and generic FEC schemes to recover from packet loss [1, 9], and some RTP payload formats supporting interleave of audio frames are presented in [10, 11]. These methods, however, do not take full advantage of the internal structure of encoded audio frames to improve the network bandwidth efficiency. On the other hand, the efficient combinations of transport and coding methods designed for video streaming, such as slice interleaving [12], are not directly applicable in the audio domain. This has been a major motivation for our work, which is summarized in this paper.

In this paper we address techniques for balancing between the contradictory requirements for high quality and interactivity in audio streaming over a packet-switched network. The system presented in this paper is built upon our previous framework [13–15]. Our framework is a combination of AAC bitstream processing for increased error robustness and network-based error robustness measures, namely, selective retransmission and priority-based redundancy addition. We have further improved our system with a better tradeoff between bandwidth efficiency and error robustness. Although we have been using the AAC codec for our system implementations, the proposed schemes could also be applied to other audio codecs following the same kind of coding principles.

This paper is organized as follows. After the introduction in Sect. 1, we outline the perceptual audio compression and compressed bitstream processing for increased error robustness in Sect. 2. Then we illustrate the principle of compressed audio data fragmentation for efficient packetization and robust transmission in Sect. 3, followed by system evaluation in Sect. 4. Following the discussions in Sect. 5, we conclude the paper in Sect. 6.

2 Perceptual audio coding and coded bitstream processing

2.1 Perceptual audio coding principles

Perceptual audio coding is the leading paradigm for compressing generic audio such as music, as opposed to speech. The best-known perceptual coders include MPEG Layer III (MP3), MPEG Advanced Audio Coding (AAC), and OggVorbis. They are all based on a time-to-frequency transform, such as the Modified Discrete Cosine Transform (MDCT), and aim at near-transparent quality and high compression ratio by removing the frequency components that are irrelevant for human perception [16]. A typical transform window length is around 2000 samples: for AAC it is 2048 for long windows (one long window per frame) or 256 for short windows (8 short windows in a frame) [17, 18]. Because the first half of each transform window overlaps with the preceding window, each AAC frame gives 1024 time domain samples as output and 1024 frequency domain samples aligning with the filtering window of the following frame.

In perceptual audio coding MDCT coefficients are usually grouped in sections. The range of possible values for spectral coefficients in each section is specified by a scalefactor. This is why the perceptual significance of scalefactors is higher than for individual spectral coefficients. There are different techniques for quantizing and coding the spectral data used in different codecs. Variable Length Coding (VLC) such as Huffman coding is often applied for scalefactors and spectral coefficients separately. In addition to the scalefactors and spectral coefficients, each audio frame usually also contains headers and flags indicating Huffman codebook indices, transform window types, and other relevant information needed by the decoder. Because the side information is mostly essential for the decoding process of a frame, we refer to it as critical data, which is perceptually the most significant part in the bitstream.

In most published standards for audio streaming, transport units (packets) contain timely consistent entities, e.g. audio frames. This is why most of the existing error concealment methods deal with complete frame loss. Such methods include muting, frame repetition, and even more advanced techniques, such as interpolation or frame replacement based on beat-pattern analysis [1,2,19–22]. However, to achieve the same perceptual quality, error concealment is typically simpler for a few missing or erroneous individual MDCT coefficients than for an entire audio frame.

2.2 Characteristics of AAC bitstreams

We have used the MPEG AAC codec for our system implementation due to the fact that AAC still represents the state of the art in standardized generic audio coding. In addition to scalefactors and quantized MDCT (QMDCT) coefficients, each AAC frame contains critical data, such as flags and data for selecting window type and Huffman codebook indices for each section [17]. Therefore, the AAC bitstream format sets certain limitations for fragmenting the data and rearranging it into transport packets. The major cause of error propagation is the Huffman coding for scalefactors and QMDCT coefficients. In addition, there can also be zero sections defined by a special zero codebook index: all MDCT coefficients in zero section are zeros and thus are not coded at all as Huffman codewords. For these reasons it is impossible to read the scalefactor and spectral data correctly without the critical data [13].

Delta Pulse-Code Modulation (DPCM) is applied to the scalefactor data before Huffman coding. In DPCM, only the first scalefactor of each frame (global gain) is stored as an original value. For the rest of these scalefactors only the difference between the current and preceding scalefactor is coded. QMDCT coefficients in AAC are also coded as Huffman codes, each codeword comprising two or four QMDCT coefficients. Thus, losing one codeword causes loss of two or four adjacent spectral samples, depending on the codebook used for the particular section. The coefficients and scalefactors are not equal in terms of priority; lower frequencies are perceptually more significant than higher frequencies in general. Because of critical flags and Huffman coding, the AAC bitstream format is extremely vulnerable to bit errors. This problem has been tackled in the MPEG-4 AAC standard by introducing optional ER tools for error resilience [17]. The ER tools allow protection of critical bits with FEC and prevention of error propagation using reversible variable-length coding (RVLC) for scalefactors and a Huffman code reordering tool for the MDCT coefficients [17]. These methods drastically improve the bit error resilience at the cost of some redundancy overhead. However, they are not effective if an entire frame of data in the form of a transport packet is lost.

2.3 Modifying AAC bitstream for increased error robustness

If the critical data of a frame are lost, the whole frame is considered lost. In this case, traditional error concealment techniques, such as repeating the previous frame, have to be used to recover the lost frame. However, using our fragmentation-based approach we can assume that in most cases there are only individual Huffman codewords missing in positions known by the decoder. Based on this assumption, we propose relatively simple yet effective strategies to conceal missing scalefactors and QMDCT coefficients.

2.3.1 Scalefactor coding

Due to DPCM coding, even one single missing scalefactor value may cause all the consequent values to be erroneous. AAC ER tools mitigate the problem using symmetric RVLC Huffman codewords instead of conventional Huffman coding [6]. When a bit error is detected, the decoder can start reading RVLC codewords from the end. This method is effective against individual bit errors or scalefactor losses, but if there are more errors or error bursts, scalefactors between the first and the last corrupted codeword cannot be recovered reliably. Therefore, the RVLC coding scheme is not an appropriate solution for recovering several separate missing scalefactors in each frame, this being the scenario in the case of packet loss with our packetization scheme.

In [15] an alternative method for coding scalefactors was proposed. We model the contour of scalefactors with a straight line as shown in Fig. 2. Huffman coding with

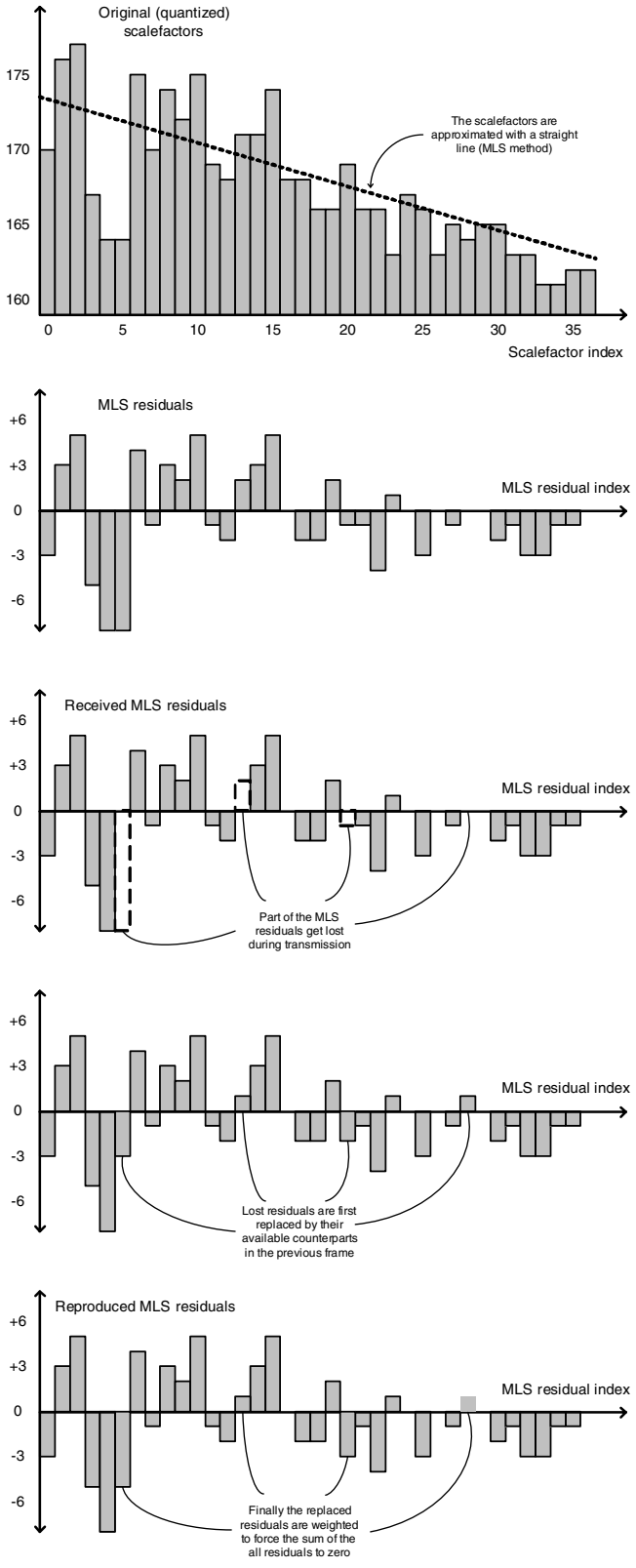


Fig. 2 The proposed scalefactor coding and recovery algorithm for the missing scalefactors illustrated

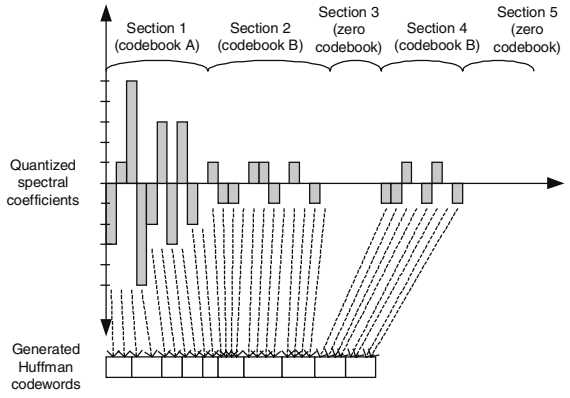
a specific codebook is then used to encode the minimum least-squares residuals. The cost of improved robustness is slightly decreased compression efficiency: in our test system, the quantized angular coefficients for the approximated values take five extra bits and the Huffman-coded residuals typically also take slightly more bits than conventionally coded DPCM values. In our tests the total frame size increased by up to 2% when the proposed coding scheme was used.

However, the proposed scheme significantly facilitates error concealment of the lost scalefactors. We have noticed that the residuals often follow the same kind of pattern in adjacent frames. This is why it is reasonable to replace a missing scalefactor with the corresponding scalefactor in the previous frame. As the residuals are known to be distributed evenly around zero due to the characteristics of the minimum least-squares method, the replaced residuals can be weighted to force the sum of the residuals to become zero. The coding and error concealment algorithms are illustrated in Fig. 2.

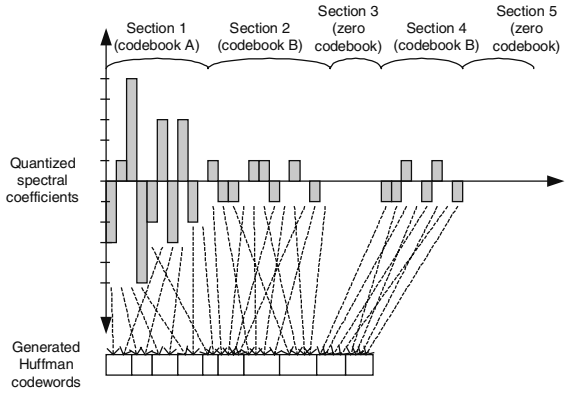
2.3.2 Quantized spectral coefficients

In the AAC bitstream each Huffman-coded spectral element represents two or four adjacent QMDCT spectral coefficients, depending on the actual codebook index. Thus, loss of one codeword means loss of two or four conjunctive spectral samples. From an error correction perspective, it is more favorable to lose separated rather than clustered spectral samples. This is why we are interleaving spectral coefficients so that each Huffman codeword contains non-adjacent coefficients. To put it simply, each section is divided into two or four parts (depending on the number of coefficients per codeword). Then each codeword is generated so that there is one coefficient from each part, according to the interleaving algorithm illustrated in Fig. 3.

The codebook index of each section sets a limit for the maximum absolute value of each QMDCT coefficient, and it is possible to predict the absolute values of the missing coefficients by interpolation or coefficient repetition. In practice, it is very difficult to predict missing QMDCT codewords reliably due to the properties of MDCT. We have tried to use some simple methods for estimating missing QMDCT values, but only very small, if any, improvements in perceived quality have been achieved in comparison to the simplest imaginable method: replacing the missing QMDCT coefficients by zeros [15]. A clearly better result can be reached by replacing every missing spectral sample by the average of the corresponding samples in the preceding and following frames in the inverse QMDCT (IQMDCT) domain. This method is shown in Fig. 4.



a) Conventional QMDCT coding in AAC.



b) Proposed QMDCT coding with interleaving

Fig. 3 Conventional and modified Huffman code generation for QMDCT coefficients

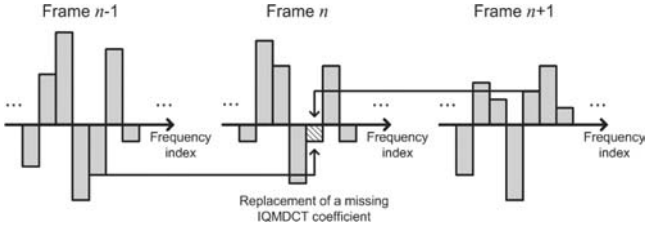


Fig. 4 Replacing an individual spectral coefficient in the inverse quantized MDCT domain with simple interpolation

3 Transport and packetization

3.1 Real-time audio transport

A fundamental issue in real-time transport of fragmented audio data is the priority distinction. As discussed above, for AAC streaming it is essential to have the critical data to decode any of the remaining data (scalefactors and spectral coefficients). This is why appropriate techniques have to be utilized to transport the critical data more reliably than the lower priority data.

Generally speaking, there are two solutions to increasing the reliability of data transport: adding redundancy to the data stream and using retransmissions. Schemes based on redundant data transport are often called forward error correction (FEC) schemes. In their most simple form the sender just replicates data elements. Since multiple copies of one data element are transmitted in different packets, the element is lost only if all copies of it are lost. This decreases

data loss probability but highly increases network bandwidth usage regardless of packet loss rates. Data replication is easy to implement for various kinds of transport channels because there is no need to change the transport protocols but only the payload format.

Retransmission-based strategies use network resources more efficiently because the data packets are replicated only if the original packet is lost. However, to use retransmissions, a feedback channel from receiver to sender is required to carry retransmission requests and extra delay is introduced in case of packet loss. This is why retransmissions can be used only if a reasonable latency can be tolerated and a feedback channel is available. However, a retransmission-based protocol is more difficult to implement than data replication. In addition, in a fixed network infrastructure packet losses are often caused by congestion. In this case, congestion control should be used in conjunction with retransmissions, which may be difficult with streaming applications. On the other hand, in wireless access networks retransmission-based transport strategies can be perfectly suitable.

RTP is the de facto standard for carrying content of a real-time nature over IP networks [23]. Originally, RTP was especially designed for multicast teleconferencing systems. Although various schemes for reliable multicast have been proposed, it is typically not reasonable to use traditional retransmissions with multicast streaming applications because multiple feedback messages and retransmissions between all the multicast group members could easily cause overwhelmingly redundant traffic and overload the network. This problem is called feedback implosion [24]. Because of the multicast support and strict real-time requirements, retransmissions were not included in the original RTP specification. However, proposals have been made to extend traditional RTP by selective retransmissions [25]. One-to-one (unicast) multimedia streaming applications can especially benefit from this feature to a great extent.

Selection of the optimal transportation and packetization strategies greatly depends on the application and the network conditions. If a multicast application, such as teleconferencing with a large number of participants or Internet radio, is involved, the optimal strategy is to avoid retransmissions and use redundancy alone to deliver critical data more reliably. In contrast, retransmission-based transport strategies may be perfectly suitable for AoD applications with more relaxed latency requirements.

3.2 Retransmission-based transport strategies

Use of data fragmentation and selective RTP retransmissions together for streaming perceptually coded audio was discussed in our earlier work [14]. In the proposed technique, data are arranged in different packets, depending on the proportional priority. Then, higher priority packets are transmitted before corresponding low priority packets. This arrangement allocates more time to retransmission attempts for the critical packets than others because the receiver has to wait for the lower priority packets before reconstructing audio frames. In addition, the network resource utilization can be efficiently controlled: if there are too many high priority retransmissions, the sender may drop some low priority packets intentionally to avoid overloading the network.

However, in streaming applications there is a definite deadline for data delivery, which limits the number of retransmission attempts. Therefore, it is possible to lose the critical section of some frame(s) in spite of retransmissions. Actually, the system described above is especially vulnerable to critical packet loss, as illustrated in Fig. 5. Whenever a critical data section of AAC data is lost, a decoder cannot read the corresponding Huffman codewords in the lower priority packets. In this case, the system cannot continue reading the Huffman coded data beyond the missing frame in any of the low priority packets. Thus, it is not just the current frame that is actually lost; all the data that belong to the following frames also becomes useless. This is illustrated in Fig. 5: frame A is read normally, but loss of the critical section for frame B also makes frame C unreadable due to loss of synchronization.

The most trivial solution for this kind of error propagation is to add delimiters or extra fields showing the length of each block. However, the extra cost of increased overhead would typically be significant. A better solution to deal with the issue is to reserve fixed-size slots for each frame and a reservoir area for codewords that do not fit in the base slot [15]. This is how the data sections for each frame can be made to start at a known position and non-readable sections

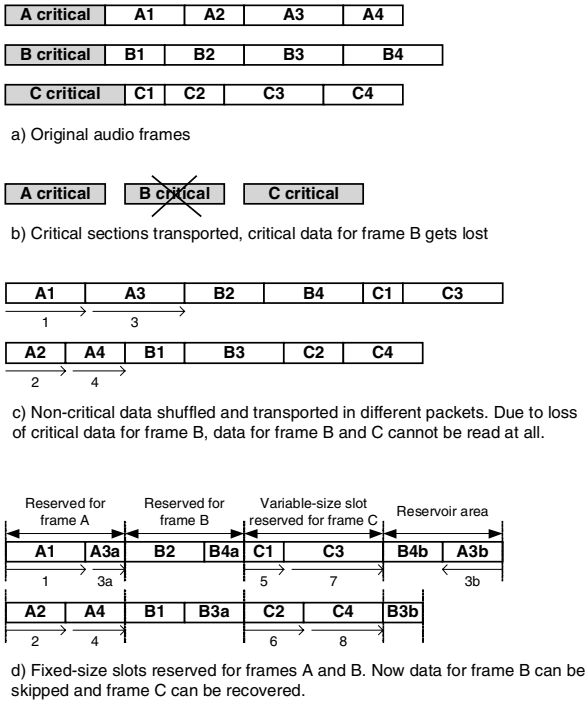


Fig. 5 Shuffling variable-length codes in multiple packets. *Numbers* denote the order in which codewords are read

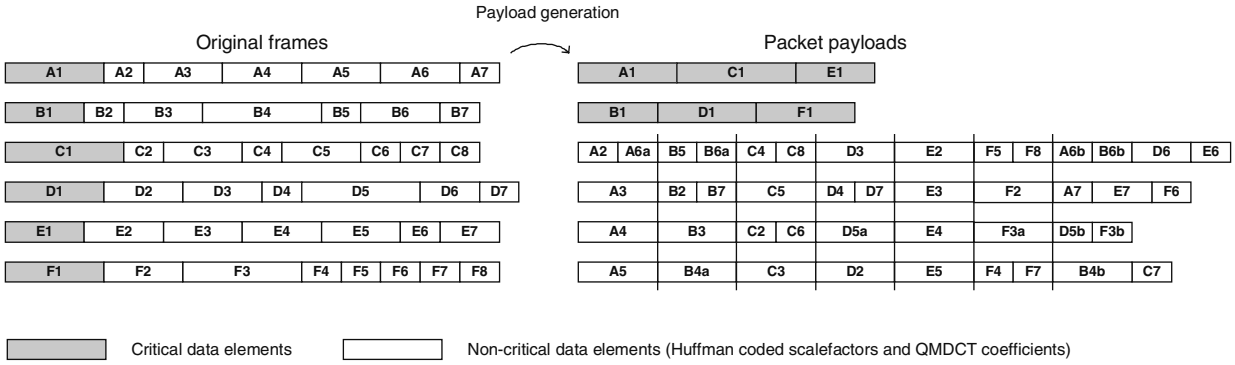


Fig. 6 Packetization scheme for retransmission-based transport system illustrated

can be skipped. In this case the synchronization is lost only between the codewords located fully or partially in the reservoir area. Because data in each frame are arranged from lowest to highest audible frequencies, the effect of losing the last codewords is not perceptually as serious as losing the first codewords. Phase d in Fig. 5 illustrates how the scheme works in practice, in contrast to the generic situation.

The packetization strategy involving two different packet priority classes and the slotting technique is illustrated in Fig. 6. The loss rate for the critical packets is assumed to be low due to retransmissions. If one of the critical packets is lost, error propagation to the other frames is restricted to the reservoir section only. Critical data sections are interleaved among the critical packets to avoid loss of adjacent packets. This allows us to employ traditional frame-based error concealment techniques to recover the missing frames.

3.3 Redundancy-based transmission strategies

Transport and packetization techniques based on added redundancy can be relatively cost efficient, especially if the proportional amount of critical data is low. This is because only the critical portion of the data has to be replicated. In our earlier work concerning audio streaming with data fragmentation and shuffling among multiple packets, the major focus was on redundancy-based transport techniques [13].

The basic principles for generating packet payloads in that scheme can be summarized as follows:

(1) All the data elements that belong to the frames of the current interleaving cycle are written in packets in the same order in which they appear.
(2) Critical data sections of one frame are written in two or more packets in parallel (adding redundancy).
(3) Lower priority data elements are shuffled among all the packets that belong to the current interleaving cycle.

Fig. 7 shows an example of payload generation when the length of the interleaving cycle is six frames, also giving six packets as output. If multiple packets are lost, the number of lost critical sections can be limited by designing rules for shuffling and adding redundancy carefully. Most importantly, no critical sections of two different frames should be written in the same set of redundant packets. It is also beneficial to use pseudorandom shuffling sequences instead of direct interleaving for the non-critical data elements. Otherwise, in the case of packet loss the distribution of lost frequency components may follow some kind of regular pattern, which can lead to more annoying perceptual experience than loss of randomly selected frequencies.

Even in this scheme it is possible that all redundant critical data elements are missing for some frames if all the packets containing the redundant data happen to get lost. In this case the error propagation applies to low priority data elements detached from the critical data. This

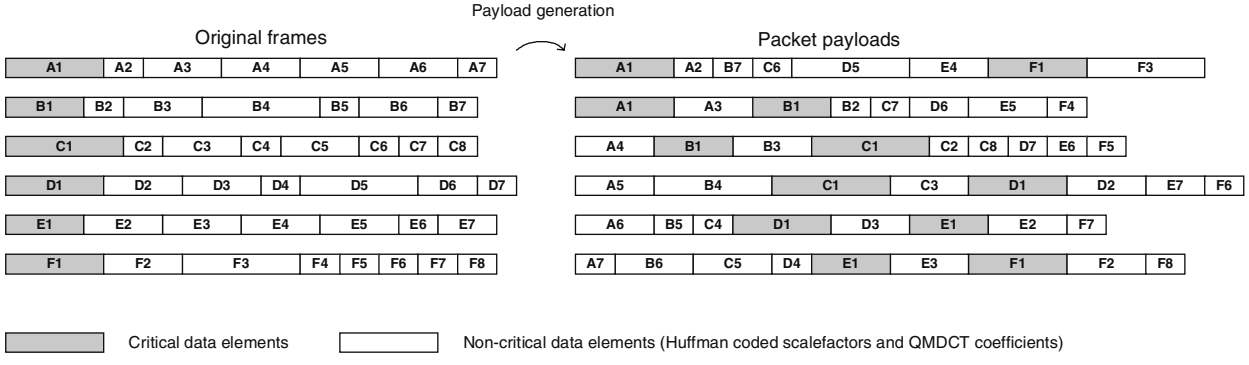


Fig. 7 Redundancy-based packetization scheme illustrated

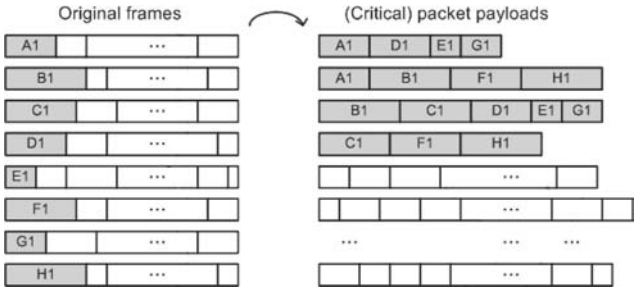


Fig. 8 Packetization scheme for a system with redundancy and retransmission-based error recovery combined

can be avoided with fixed-size slots and reservoir areas for non-critical codewords, just like in the retransmission-based packetization scheme. For simplicity, this is not illustrated in Fig. 6. It should be noted that this technique is needed only for the frames with no related critical data in the same packet – when the critical data are present, the system is guaranteed to be able to read the non-critical elements in that packet correctly.

3.4 Hybrid strategies

One possibility for tradeoff between redundancy overhead and retransmission delay is to use both retransmissions and redundancy-based error recovery schemes in parallel. In this paper we consider only simple repetition to protect the critical sections. This alternative uses a packetization scheme similar to the retransmission-based transport system illustrated in Fig. 6. The only difference is that each critical section is repeated in two different critical packets. The basic principle of generating critical packets in this scheme is shown in Fig. 8.

4 Performance evaluation

We have implemented both the retransmission-based and redundancy-based packetization mechanisms in a streaming software application to test and evaluate different approaches to error concealment, packetization, and transport. The AAC bitstream parser and decoder used for payload generation and frame reconstruction as well as decoding is able to parse MPEG-2 AAC bitstreams with main or low complexity (LC) profiles. AAC stereo bitstreams representing different music styles and using a 44.1-kHz sample rate and a 128-kbit/s encoded bit rate were used for primary tests. RTP was used for real-time data transport and RTCP for carrying retransmission requests when the selective retransmissions were utilized.

4.1 Listening tests

To rationalize the basic concept of fragmenting audio frames and spreading the scalefactors and spectral coefficients of one frame into several different transport units, we also performed a formal listening test to compare the subjective performance of the proposed error robustness measures against a simple frame repetition method. In the frame repetition method each missing frame is replaced with the previous correctly received frame.

Table 1 Definition and description of programme material used in listening test

Programme	Description
Country	Female lead vocal, strummed acoustic and solo slide guitars, bass, drums and percussion
Ballad	Slow rock ballad with lead guitar part and overall reverberant mix; no vocals
Rock	Highly compressed rock anthem, male lead vocal
Jazz	Female lead vocal with clean electric guitar backing only
Classical	Church recording of 'Herr, Unser Herrscher' from J.S.Bach's 'St. John Passion'
Dance	Up-tempo, repetitive drum, bass and organ dance groove; no vocals

4.1.1 Test stimuli

Of course, there are more advanced frame-based error concealment methods than simple frame repetition. However, it is a very simple method that provides relatively good subjective performance with many different kinds of audio. To make the comparison fair, the error concealment in the fragmentation-based scheme is kept simple as well. RTP payloads were produced using scalefactor coding based on linear approximation and double redundancy for critical data (each original critical data section was repeated twice in different packets).

Six AAC bitstreams representing a range of popular music styles were used for the tests. Table 1 shows the naming scheme of each musical programme and gives a brief description of its content. Each programme was about 30 s in duration. Test stimuli were generated using our streaming software that is configurable to use either a traditional streaming mode with frame repetition or a fragmentation-based streaming mode. Packet losses were simulated by using a random number generator to decide whether to send a packet or not. In practical networks, packet losses tend to be bursty, but the burstiness can be smoothed with our interleaving scheme. In our system the depth of the interleaving cycle is 64 frames. Typical packet loss burst length is substantially smaller, which makes use of more complex packet loss models unnecessary. Three different theoretical packet loss rates were used – 10, 20 and 30% lost packets.

If the actual packet loss rate was shown to have deviated substantially from the theoretical packet loss rate, the sample was reproduced to guarantee equitable comparison.

4.1.2 Test design

A forced-choice binary paired comparison design was implemented with full permutation pairs (A–B and B–A)

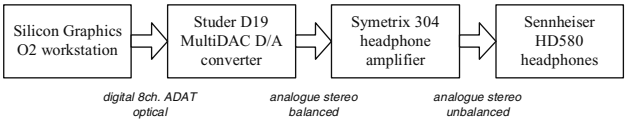


Fig. 9 Packetization scheme for a system with redundancy and retransmission-based error recovery combined

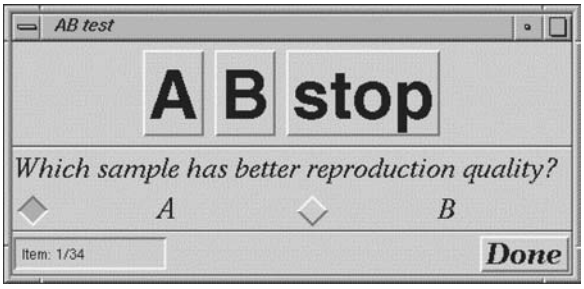


Fig. 10 Test administration user interface

presented between both of the error concealment schemes. Comparisons were only made between stimuli having the same packet loss rates. Thus, for all presentation permutations, packet loss rates and programmes resulted in, respectively, $2 \times 3 \times 6 = 36$ presentation pairs. Twelve training pairs that were representative of the range of the stimuli included in the test were presented to the listeners to begin with as training in the type of auditory attributes that they would be required to grade as well as familiarizing them with the grading user interface. These training results were later discarded from the analysis.

Six listeners who had experience and had previously shown expertise in performing listening tests on error concealment algorithms were chosen to perform the tests. None had any hearing loss, and all were males between 20 and 30 years of age.

The test was administered using the Guineapig listening test system [26] in a controlled, silent listening environment, specified in [27]. The audio signal chain for presentation is shown in Fig. 9, and the user interface used for presentation of pairs is shown in Fig. 10. The stimuli were 16-bit, 44.1-kHz PCM recordings of the decoded material. The stimuli were presented over headphones. All stimuli were loudness aligned using Moore's steady-state loudness model [28] to be 20 sones when averaged across the entire sample. This alignment was performed to negate any biasing effect associated with the loudness of one error recovery method over another.

4.1.3 Test results and analysis

The proportions of preference scores for all factors are shown in Fig. 11 and the results of binomial tests between the two algorithms, along with the proportions in numerical format, are shown in Table 2. Most test cases show that the fragmentation-based packet loss recovery scheme was preferred to the frame repetition method. At a 10% packet loss rate, however, the difference is not as clear as at higher loss

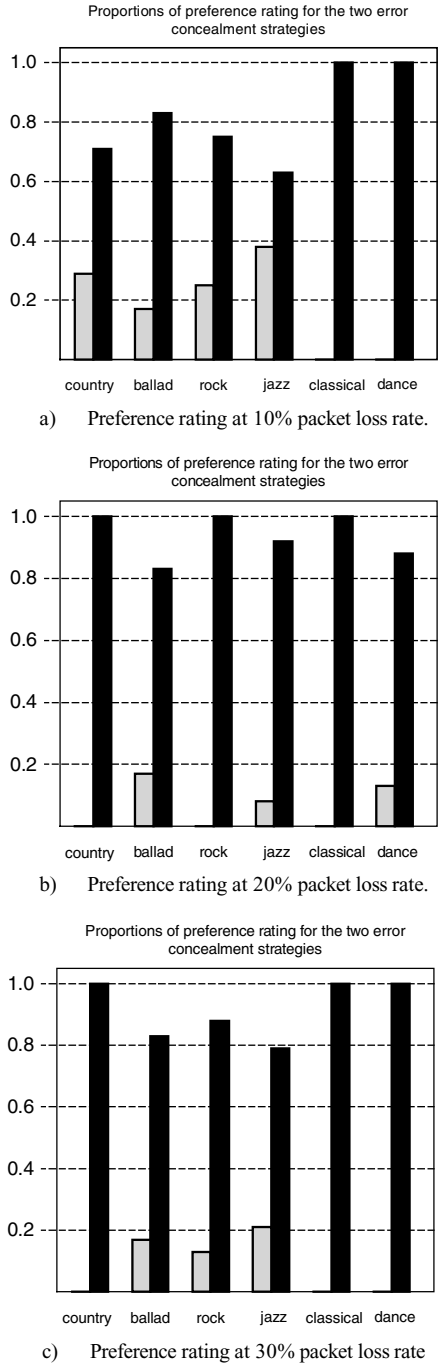


Fig. 11 Proportions of preference rating for the two approaches, fragmentation-based (*solid bar*) and frame-repetition-based (*grey bar*)

rates. In 15 of the total 18 test cases, there is a statistically significant difference at the 0.01 level in preference rating.

At low packet loss rate the fragmentation-based scheme and frame repetition both work relatively well. This is why the variance in preferences is the highest at the lowest packet loss rate. Anyway, in all cases the perceived distortion is different by nature, depending on which method is used for error concealment. Frame repetition causes beat doubling,

Table 2 Results of binomial significance tests between the two algorithms. Asterisks denote significant preference for the fragmentation-based algorithm. No results show significant preference for the frame repetition at the 5% level (e.g. $P \geq .950$)

		Programme					
		Country	Ballad	Rock	Jazz	Classical	Dance
Packet loss	10%	.064	.002**	.023*	.307	.000**	.000**
	20%	.000**	.002**	.000**	.000**	.000**	.000**
	30%	.000**	.002**	.000**	.007**	.000**	.000**

* Significant difference at the .05 level
** Significant difference at the .01 level

echoes and clicks, whereas the fragmentation-based method makes some frequencies vanish occasionally. This causes irregular smooth bubbling or frequency-shifting artefacts. The results clearly indicate the advantage of using compressed domain error concealment for individual data elements instead of simple frame-based error concealment in general, over a wide range of musical styles. Especially at rather high data loss rates, data fragmentation can significantly improve the subjective performance of error concealment. Because only simple schemes are used for error concealment, test cases are comparable in terms of computational cost.

4.2 Network resource utilization

Regarding network utilization, the retransmission-based transport strategies behave very differently in comparison to redundancy-based error recovery techniques. When a redundancy-based system is used, the average transmission rate remains constant even if the packet loss rate varies. Of course, in this case the bit rate also contains the redundancy overhead. If simple critical data replication is applied, the average residual frame loss rate p_{fl} and redundancy overhead T_o can be estimated by Eqs. 1 and 2, respectively:

$$p_{\mathrm{fl}} = p^{r+1}, \tag{1}$$

$$T_o = cr\,, \tag{2}$$

where p is the packet loss rate, r is the number of critical data section replicates and c is the proportional amount of critical data. At low packet loss rates (below 10%) the frame loss rate gives a good idea of reproduction quality as well.

By comparison, the retransmission-based approaches result in lighter network resource use when packet loss rate is low, but heavier use of the network resources due to retransmission overhead when there are more lost packets. If packet losses are caused by congestion, as is typical in fixed IP networks, retransmissions may even impair the network performance experienced by other users. However, the proposed selective retransmission scheme that allocates most retransmissions for packets of highest priority can be used to significantly reduce the retransmission overhead and still maintain reasonable quality. A more detailed analysis about the performance of the system in terms of network resource use, including simulation results, is described in [14].

If there is no redundancy in critical packets, the residual frame error rate in the retransmission-based scheme depends on the maximum number of retransmission attempts allocated for the critical packets, n. We expect that the packet error rate will be constant (p) for both upstream and downstream directions and the retransmissions are based on negative acknowledgements (NACKs). The initial media packet is lost at probability p. Retransmission fails in two cases: either the NACK message is lost (probability p) or the NACK message is received, but the retransmitted media packet is lost at probability $p(1-p)$; therefore each retransmission attempt fails at probability $p + p(1-p) = 2p - p^2$. Combining these cases, the theoretical residual packet loss rate follows Eq. 3.

Retransmission causes overhead only if the NACK message is received (probability $1-p$). Considering the residual frame loss rate after each retransmission attempt derived above, the corresponding retransmission overhead T_o can be calculated by Eq. 4. In these computations we expect that NACK-based retransmission schemes will be preferred for real-time communications because the retransmission delay can be minimized as the receiver can send a NACK message immediately when a gap in the received packets' sequence numbers is detected:

$$p_{\mathrm{fl}} = p(2p - p^2)^n \tag{3}$$

$$T_{rt} = \sum_{i=1}^{n} cp(1-p)(2p-p^2)^{i-1}\,. \tag{4}$$

If one replicate for each critical section is added to another critical packet (single redundancy), the equations become Eqs. 5 and 6:

$$p_{\mathrm{fl}} = p^2(2p - p^2)^n \tag{5}$$

$$T_{o+rt} = c + \sum_{i=1}^{n} cp^2(1-p)(2p-p^2)^{i-1}\,. \tag{6}$$

As an example, the theoretical residual frame loss rate and total downstream overhead for different cases are illustrated in Fig. 12. The proportional amount of critical data is assumed to be 10%. This is a realistic assumption for a perceptually coded audio track with CD or near-CD quality. At lower bitrates, however, the proportion of critical data is usually higher.

5 Discussion

As Fig. 12 shows, network use is significantly lower for a retransmission-based streaming system in comparison to redundancy-based strategies. Even if the feedback traffic were to be taken into account, the network load would not grow radically. Selective retransmissions allow use of the saved bandwidth for retransmitting lower priority

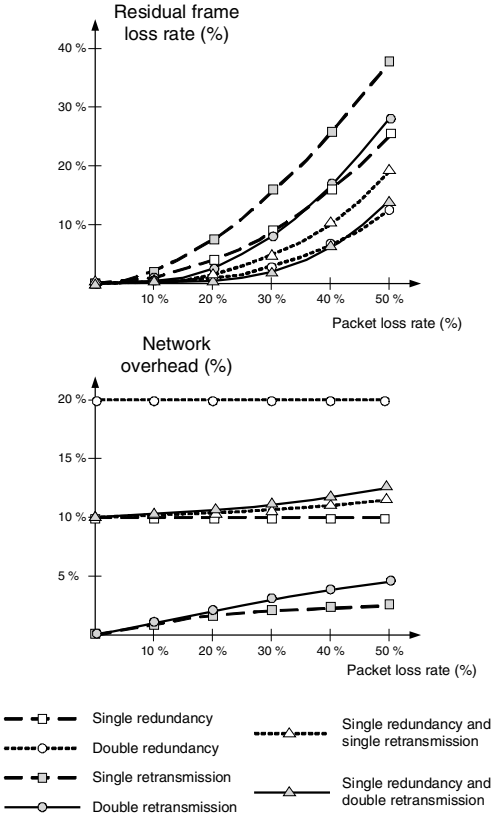


Fig. 12 Frame loss rate and network overhead with different packetization schemes

data also, which makes the scheme very flexible. This proposed scheme could be especially suitable for wireless communications.

However, a retransmission-based system is not fully applicable in all situations. Usage of simple NACK-based retransmissions without congestion control cannot be highly recommended for traditional IP networks, where the majority of packet losses indicate congestion. In these kinds of circumstances, any retransmission may degrade the network performance even more, also influencing the other users of the network. In a multicast environment, retransmissions may be even more harmful because there are a lot of communicating parties involved sending and receiving feedback messages and retransmitted packets.

When critical sections from multiple audio frames are packed in a single packet, one missing critical packet makes multiple frames near to each other become lost. This typically causes more severe subjective distortion in reproduction quality than for the same frame loss rate in traditional frame-based streaming with smooth packet loss distribution. Therefore, a hybrid solution (single redundancy for critical data and selective retransmissions together) is a highly beneficial scheme for unicast audio streaming applications with high quality requirements in difficult network conditions.

The optimal number of frames per shuffling or interleaving cycle is another issue to be considered. A long sequence is more robust against bursty packet loss and allows more time for critical packet retransmissions. On the other hand, it causes long latency due to interleaving and deinterleaving delays and requires a large buffer at the receiver. Thus, a very long sequence is not an option in highly interactive communications or client devices with limited memory and processing power. To get the best out of the scheme, the application should allow total latency of at least 5 s. This is acceptable for classical streaming, Internet radio and many multimedia broadcasting applications, but not traditional telephony, which requires a total end-to-end delay lower than 0.5 s.

In general, the proposed schemes are efficient in terms of computational complexity and memory consumption. Audio frames can be stored in the interleaving buffer in compressed format. Therefore, the required interleaving buffer size is reasonable. The proposed scalefactor coding and error concealment methods are based on simple algorithms, and there are typically only few missing scalefactors and spectral coefficients in each frame. This keeps the processing overhead low.

In a wireless access network, packet size optimization often plays a significant role because large packets are more likely to be hit by bit errors in the radio channel than small packets. Traditionally, arbitrary fragmentation of audio frames is not allowed because loss of one fragment would render the related fragments, and thus the whole frame, useless. However, the fragmentation scheme presented in this paper is more flexible. By selecting the number of packets with different priorities per cycle appropriately, it is possible to make the critical packets smaller than the other packets, which can facilitate the delivery of the high priority packets over a wireless channel.

It is noteworthy that the AAC specifications define only the bitstream format and decoder functionality. Different encoders may have dissimilar approaches for generating an AAC-compliant bitstream. For example, the proportion of bits allocated for scalefactors and QMDCT data may vary from encoder to encoder, even if the original sample is the same. Due to the variance in encoding strategies, the compressed domain error concealment methods can perform differently when AAC bitstreams generated by various encoders are tested. However, we believe that the observations reported in this paper would be compatible with other mainstream codec implementations. It would also be possible to optimize existing encoders to support the proposed schemes.

6 Conclusions

The traditional paradigm in real-time multimedia streaming is to use unreliable transport protocols to carry homogeneous data units representing multimedia content clips in a timely, continuous order. This paper presents an alternative approach that fragments individual frames into smaller data segments with different perceptual significance. This fragmentation enables different packetization schemes for robust data transportation. Three different packetization and transport schemes are summarized in this paper; these schems

increase error robustness by employing selective retransmissions, priority-based added redundancy or a mixture of these two. Furthermore, we have proposed modifications to the baseline AAC bitstream format to make it more suitable for error-robust transport.

The proposed approach facilitates compressed domain error concealment because each packet loss erases only individual spectral components spread in several frames. The proposed scheme also allows efficient uneven error control via added redundancy or selective retransmissions because data components of different proportional priority are allocated in different packets. The rationale behind the scheme has been verified by theoretical analysis of network performance and formal listening tests comparing the subjective performance of the proposed techniques against systems using traditional frame-based error concealment.

Our test results show that the error concealment methods based on the proposed approach improve audio quality in comparison to the traditional frame-based error concealment when packet losses occur. The extra cost of increased network resource consumption caused by data retransmissions or added redundancy can be kept reasonably low due to data prioritization. In this way, more retransmission attempts or duplicated data sections can be allocated for the most critical parts of data.

References

1. Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.-C., Vega-Garcia, A., Fosse-Parisis, S.: RTP payload for redundant audio data. IETF RFC 2198 (1997)
2. Wah, B.W., Su, X., Lin, D.: A survey of error-concealment schemes for real-time audio and video transmissions over the Internet. In: Proceedings of the IEEE International Symposium on Multimedia Software Engineering (MSE '00), pp. 17–24. Taipei, Taiwan (2000)
3. Lauber, P., Sperschneider, R.: Error concealment for compressed digital audio. Convention Paper 5460 at the 111th AES Convention, New York (2001)
4. Miao, L., Lu, J., Gu, J.: An improved error resilience scheme for transmission of MPEG-4 audio over EGPRS. In: Proceedings of the IEEE Vehicular Technology Conference (VTC Fall '01), pp. 414–417. Atlantic City, NJ (2001)
5. Sperschneider, R., Homm, D., Chambat, L.-H.: Error resilient source coding with variable-length codes and its application to MPEG advanced audio coding. Audio Engineering Society Convention Paper 5271 at the 109th AES Convention. Los Angeles (2000)
6. Sperschneider, R., Homm, D., Chambat, L.-H.: Error resilient source coding with differential variable-length codes and its application to MPEG advanced audio coding. Audio Engineering Society Convention Paper 5555 at the 112th AES Convention. Munich, Germany (2002)
7. Chawla, K., Driessen, P., Qiu, X.: Transmission of streaming data over an EGPRS wireless network. In: Proceedings of IEEE Vehicular Technology Conference (VTC '00), vol. 1, pp. 118–122. Tokyo (2000)
8. Transparent End-to-End Packet Switched Streaming Service (PSS): RTP usage model. 3rd Generation Partnership Project TR 26.937 V6.0.0 (2004)
9. Rosenberg, J., Schultzrinne, H.: An RTP payload format for generic forward error correction. IETF RFC 2733 (1999)
10. Finlayson, R.: A more loss-tolerant RTP payload format for MP3 audio. IETF RFC 3119 (2001)
11. van der Meer, J., Mackie, D., Swaminathan, V., Singer, D., Singer, P.: RTP payload format for transport of MPEG-4 elementary streams. IETF RFC 3640 (2003)
12. Stockhammer, T., Viegand, T., Oelbaum, T., Obermeier, F.: Video coding and transport layer techniques for H.264/AVC-based transmission over packet-lossy networks. In: Proceedings of the International Conference on Image Processing (ICIP '03), pp. 481–484. Barcelona, Spain (2003)
13. Korhonen, J.: Error robustness scheme for perceptually coded audio based on interframe shuffling of samples. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '02), pp. 2053–2056. Orlando, FL (2002)
14. Korhonen, J.: Robust audio streaming over lossy packet-switched networks. In: Proceedings of International Conference on Information Networking (ICOIN '03), pp. 1343–1352. Jeju Island, South Korea, (2003)
15. Korhonen, J., Wang, Y.: Schemes for error resilient streaming of perceptually coded audio. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP '02), vol. 5, pp. 740–743. Hong Kong (2003)
16. Painter, T., Spanias, A.: Perceptual coding of digital audio. Proc. IEEE **88**(4), 451–515 (2000)
17. Coding of Audio-Visual Objects – Part 3: Audio. ISO/IEC International Standard 14496-3 (2001)
18. Wang, Y., Vilermo, M.: Modified discrete cosine transform: its implications for audio coding and error concealment. J. Audio Eng. Soc. **51**(1/2) (2003)
19. Herre, J., Eberlein, E.: Error concealment in the spectral domain. Convention Paper 3364 at the 93rd AES Convention. San Francisco, USA (1992)
20. Quackenbush, S., Driessen, P.: Error mitigation in MPEG-4 audio packet communication systems. Audio Engineering Society Convention Paper 5981 at the 109th AES Convention. New York, USA (2003)
21. Wang, Y., Streich, S.: A drumbeat-pattern based error concealment method for music streaming applications. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP '02), pp. 2817–2820. Orlando, FL (2002)
22. Kauppinen, I., Roth, K.: Audio signal extrapolation—theory and applications. In: Proceedings of the 5th Conference on Digital Audio Effects, pp. 105–110. Hamburg, Germany (2002)
23. Schultzrinne, H., Casner, S., Frederick, R., Jacobson, V.: A transport protocol for real-time applications. IETF RFC 3550 (2003)
24. Li, V., Zaichen, Z.: Internet multicast routing and transport control protocols. Proc. IEEE **90**(3), 360–391 (2002)
25. Rey, L., Leon, D., Miyazaki, A., Varsa, V., Hakenberg, R.: RTP retransmission payload format. Internet draft, March 2004 (work in progress)
26. Hynninen, J., Zacharov, N.: Guineapig—a generic subjective test system for multichannel audio. Audio Engineering Society Convention Paper 4871 at the 106th AES Convention. Munich, Germany (1999)
27. Kylliäinen, M., Helimäki, H., Zacharov, N., Cozens, J.: Compact high performance listening spaces. In: Proceedings of Euronoise. Naples, Italy (2003)
28. Moore, B.C.J., Glasberg, B.R., Baer, T.: A model for the prediction of thresholds, loudness and partial loudness. J. Audio Eng. Soc. **45**(4), 224–240 (1997)