TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 597 • Publication 597

Jari Korhonen

New Methods for Robust Audio Streaming in a Wireless Environment

Tampere 2006

Tampereen teknillinen yliopisto. Julkaisu 597
Tampere University of Technology. Publication 597

Jari Korhonen

New Methods for Robust Audio Streaming in a Wireless Environment

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 19th of May 2006, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2006

ISBN 952-15-1593-7 (printed)
ISBN 952-15-1733-6 (PDF)
ISSN 1459-2045

Abstract

The rapid development of mobile computing is turning mobile terminals into fully equipped entertainment systems capable of reproducing live audio and video. However, wireless access networks still pose significant limitations for the network capacity and the quality of service experienced by the end users, compared to the high standard of service provided by the modern fixed Internet Protocol (IP) networks and access technologies based on digital subscriber lines. This dissertation concentrates in application layer solutions for optimizing the network resource utilization and audio reproduction quality in the audio streaming applications used in wireless environments. Although the major focus is in the audio streaming, many of the proposed approaches are also applicable for other types of streaming data, such as digital video and animation.

The first part of this dissertation concentrates in an audio streaming system that is based on shuffling of frequency components and critical blocks within each frame among several transport packets to achieve higher robustness against packet losses. This approach allows efficient co-design with different kinds error recovery schemes, as different level of error protection may be applied to separate frame components, depending on their priorities. We propose several alternatives for error recovery, including Forward Error Correction (FEC), selective retransmissions and a hybrid of these two strategies. Unfortunately, the state-of-the-art audio coding standards, especially Advanced Audio Coding (AAC), do not intrinsically support this kind of fragmentation and data prioritization schemes. This is why we propose also modifications to the baseline AAC bitstream format to support the suggested transport and error recovery strategies better.

In the second part of this dissertation, we focus on the characteristics of a wireless link. Several recent studies show that the wireless network resource utilization could be significantly improved if the packets containing bit errors were relayed up to the application instead of using link layer retransmissions or strong FEC included in many wireless standards. In this case, the application must be able to cope with bit errors in the payload. For this purpose, we propose a bit-error robust packetization scheme for AAC streaming. We have also studied the possibility to select adaptively between different error recovery strategies, such as partial retransmissions and application layer FEC, depending on the distribution of bit errors. However, many existing wireless technologies do not allow user to switch off the link layer error recovery mechanisms. Even in this case, a proper packetization scheme at the application layer may be beneficial to optimize the network performance. Especially packet size optimization could significantly improve application layer quality, efficiency of the wireless link resource usage and power efficiency altogether.

The proposed new methods and observations have clear potential implications to the future solutions in the field of wireless multimedia streaming. For example, the concept of prioritized packetization could be highly useful for streaming in the networks with intelligent Quality of Service (QoS) mechanisms, peer-to-peer streaming with link dispersion, and energy efficient streaming relying on bursty transmission mode. On the other hand, the proposed application layer adaptation schemes for bit-error prone environments may be proven beneficial for the cross-layer network system architectures in the future.

Preface

The work for this dissertation has been carried out at the Nokia Research Center (NRC), Audio-Visual Systems laboratory (formerly Speech and Audio Systems laboratory), Tampere, Finland, during the period from March 2001 to April 2004, and at the National University of Singapore (NUS), School of Computing, Singapore, during the period from May 2004 to April 2005.

My first thanks are due to the supervisor of my thesis, Prof. Jarmo Harju from Tampere University of Technology, for his guidance and support during this work. I also thank my supervisors at NRC, Mr. Mauri Väänänen and Mr. Jari Hagqvist, for their support in my efforts to combine work and study, and Prof. Ye Wang for arranging me the opportunity to spend a year as a visiting researcher at NUS.

Many thanks are also due to several colleagues for co-operation and fruitful discussions. I thank particularly Bogdan Moldoveanu from NRC for helping me to get started with the topic; Roope Järvinen and David Isherwood from NRC for their co-operation in the research work; Miikka Vilermo and Kalervo Kontola from NRC, and Prof. Mun Choon Chan, Prof. Wei Tsang Ooi and Yicheng Huang from NUS for inspiring discussions, and their helpful comments and suggestions.

Last but not least, I would like to thank my parents Reijo and Aulikki, and my brother Juha for all the emotional support and encouragement during the writing of this thesis.

Tampere, February 2006

Contents

List of Publications

This dissertation includes the following publications.

[P1] J. Korhonen, *Error Robustness Scheme for Perceptually Coded Audio based on Interframe Shuffling of Samples*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ‘02), Orlando, Florida, USA, May 2002, pp. 2817-2820.

[P2] J. Korhonen, *Robust Audio Streaming over Lossy Packet-Switched Networks,* Proceedings of the International Conference on Information Networking (ICOIN ‘03), Jeju Island, South Korea, February 2003, pp. 1343-1352. Reprinted in ICOIN 2003 Revised Selected Papers, LNCS 2662, Springer-Verlag, Berlin, pp. 386-395, 2003.

[P3] J. Korhonen, and Y. Wang, *Schemes for Error Resilient Streaming of Perceptually Coded Audio*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ‘03), Hong Kong, April 2003, vol. 5, pp. 740-743. Reprinted in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ‘03), Baltimore, Maryland, USA, July 2003, vol. 3, pp. 165-168.

[P4] J. Korhonen, and R. Järvinen, *Packetization Scheme for Streaming High-Quality Audio over Wireless Links*, Proceedings of the Workshop on Multimedia Interactive Protocols and Systems (MIPS ‘03), LNCS 2899, Naples, Italy, November 2003, pp. 42-53.

[P5] J. Korhonen, *Adaptive Multimedia Streaming for Heterogeneous Networks*, Proceedings of the Conference on Wired/Wireless Internet Communications (WWIC ‘04), LNCS 2957, Frankfurt Oder, Germany, February 2004, pp. 248-259.

[P6] J. Korhonen, Y. Wang, and D. Isherwood, *Towards Bandwidth Efficient and Error Robust Audio Streaming over Lossy Packet Networks*, Multimedia Systems, vol. 10, no. 5, August 2005, pp. 402-412.

[P7] J. Korhonen, and Y. Wang, *Effect of Packet Size on Loss Rate and Delay in Wireless Links*, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’05), New Orleans, Louisiana, USA, March 2005, pp. 1608-1613.

[P8] J. Korhonen, and Y. Wang, *Power-Efficient Streaming for Mobile Terminals*, Proceedings of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV ‘05), Stevenson, Washington, USA, June 2005, pp. 39-44.

List of Abbreviations

3G	3rd Generation Mobile Communications
3GPP	3rd Generation Partnership Project
A/V	Audio/Visual
AAC	MPEG Advanced Audio Coding
AC-2	Audio Coder version 2
ACK	ACKnowledgement (in context of protocols)
AMR	Adaptive MultiRate speech coder
AMR-WB	Adaptive MultiRate WideBand speech coder
AP	Access Point (in context of WLAN)
API	Applications Programming Interface
ARPANET	Advanced Research Projects Agency Network
ARQ	Automatic Repeat reQuest
AVC	MPEG Advanced Video Coding
BS	Base Station (in context of cellular telecommunications)
CCITT	Comité Consultatif International de Téléphonie et de Télégraphie
CD	Compact Disc
codec	coder/decoder
CPU	Central Processing Unit
dB	Decibels
DCCP	Datagram Congestion Control Protocol
DiffServ	Differentiated Service
DPCM	Differential Pulse Code Modulation
DVB-H	Digital Video Broadcasting: Handhelds
ER	Error Resilience (in context of MPEG)
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
FFT	Fast Fourier Transformation
FGS	Fine Grain Scalability
FR	Full Rate (in context of GSM)
GSM	Global System for Mobile communication, a standard for digital cellular mobile telecommunications
HCR	Huffman Code Reordering
HR	Half Rate (in context of GSM)
Hz	Hertz
IEEE	The Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Service
IP	Internet Protocol
ITU-T	International Telecommunication Union, Telecommunication Standardization sector
LAN	Local Area Network
MAC	Medium Access Control
MDC	Multiple Description Coding
MDCT	Modified Discrete Cosine Transformation
MLS	Minimum Least Squares
MP3	MPEG-1/2 Audio Layer III
MPEG	Moving Picture Experts Group

NAK	Negative AcKnowledgement (in context of protocols)
NAL	Network Abstraction Layer
PAC	Perceptual Audio Coder
PC	Personal Computer
PCM	Pulse Code Modulation
QMDCT	Quantized Modified Discrete Cosine Transform
QoS	Quality of Service
RFC	Request for Comment
RR	Receiver Report (in context of RTP)
RSVP	Resource reSerVation Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round-Trip Time
RVLC	Reversible Variable Length Coding
SDP	Session Description Protocol
SR	Sender Report (in context of RTP)
TCP	Transmission Control Protocol, a reliable transport layer protocol used in TCP/IP networks
TwinVQ	Transform domain Weighted Interleave Vector Quantization
UDP	User Datagram Protocol
UDP Lite	User Datagram Protocol Lite
UEP	Unequal Error Protection
VHS	Video Home System
VLS	Varable Length Coding
VoD	Video-on-Demand
VoIP	Voice over Internet Protocol, the telephony services using IP networks for speech transmission
XOR	eXclusive OR binary operation
WLAN	Wireless Local Area Network
WMA	Windows Media Audio

Chapter 1 Introduction

During the past few years, the high-speed Internet connections have spread from companies and institutions to the homes of ordinary people. Different useful and entertaining services relying on audiovisual content distribution are getting more popular. Concurrently, the Internet is going wireless. In the near future, people will expect to be using same kind of services with their mobile terminals as they are using with their home computers today.

However, the available bit rates and the stability of cellular networks are still substantially lower than in the wired Internet Protocol (IP) networks. Therefore, it is essential to tailor the wireless applications to cope with the special characteristics of the mobile media. This is true especially for demanding applications, such as multimedia streaming. This dissertation addresses some of the technical issues and challenges in the field of wireless multimedia streaming and proposes solutions for them.

1.1 Motivation and Background

The idea of IP networking is to glue data networks using different underlying technologies together into an extensive network of interconnected subnets, the Internet. According to the original Internet paradigm, the complexity of the system should lie primarily in the user devices; routers and bridges are supposed to be as simple as possible. Above the IP layer there are different transport and application layer protocols fulfilling the disparate requirements of different applications.

Increasing demand for multimedia communications over IP has encouraged development of network architectures supporting traffic prioritization, often referred as Quality of Service (QoS). There are several mechanisms for supporting QoS concepts: for example, routers can forward priority packets before other packets and a certain portion of the link bandwidth can be reserved for real-time applications. However, QoS poses also problems. Most importantly, it breaks down the original Internet design philosophy with dumb routers and smart user terminals. There are still open technical questions, such as fair signaling and resource reservation policies as well as mapping the user requirements into actual QoS parameters in different systems. It is also a huge effort to update all the network devices in the Internet to provide support for end-to-end QoS.

Even in the optimistic scenario, the best-effort IP networks will prevail for a long time. However, there are several techniques that can be used at the application and transport layer to optimize the performance of streaming applications. For example, interleaving, Forward Error Correction (FEC) or even application specific retransmission schemes can be used to combat packet losses. It is also possible to implement codec-dependent rate adaptation mechanisms to enable congestion control for the real-time traffic. Even deployment of QoS does not remove the need for well-behaving transport and application layer protocols.

1.2 Outline and Objectives of the Thesis

This thesis focuses in the methods for balancing between the network resource utilization, end-to-end transport delay, power consumption and quality experienced by the user, via application layer optimization of audio streaming. The work can be divided in two separate, although slightly overlapping modules. In the first module, improvements for AAC audio coding and the relevant transport mechanisms have been proposed and evaluated. The second task widens the scope to wireless multimedia streaming in general via a study about network characteristics in a wireless media and system level proposals to support adaptive multimedia streaming in a wireless environment.

It was realized that the mainstream generic audio coding standards and IETF transport protocols as such do not provide us with optimal support for real-time transport of high quality audio. This is the motivation for developing a streaming system with improved AAC coding and selective RTP retransmissions. In a wireless system, new challenges are faced. Especially, problems in end-to-end packet transmission are often related to the wireless transport medium itself, in contrast to the congestion related variations in the jitter and packet loss characteristics in the traditional fixed networks. In this case, appropriate application and transport layer coding and transmission strategies could significantly facilitate error recovery and improve wireless channel resource utilization and power efficiency even without accurate knowledge on the link and physical layer parameters and conditions.

In this study the proposed techniques has been evaluated both theoretically and experimentally. In the practical experiments, streaming test applications have been implemented and their performance is evaluated on real-life platforms rather than simulation environments. Major focus in the practical experiments is the validation of the theoretically derived hypothesis and studying the physical and link layer characteristics from the observations on the application layer.

Chapter 2 outlines the conceptual framework of this thesis with introduction to the network protocols and multimedia coding technologies relevant for multimedia streaming. Chapter 3 presents the proposed audio coding and transport schemes for high quality audio streaming, and outlines the framework for using the schemes efficiently. Chapter 4 addresses the issues related to adaptive multimedia streaming in a wireless environment. Chapter 5 summarizes the included publications and author's contributions to them. Finally, Chapter 6 concludes the thesis and outlines the future research directions and objectives in the area of this thesis.

Chapter 2 Fundamentals of Multimedia Streaming

In this Chapter, the conceptual framework for multimedia streaming is outlined as seen relevant for the thesis. First, the concept and the historical perspective to streaming are studied. Second, the appropriate network protocols and technologies as well as multimedia coding principles are reviewed. Although the scope of this thesis is primarily in audio streaming, many of the addressed issues are valid also for video streaming. This is why video coding is also addressed cursorily.

2.1 Scope and Definition of Streaming

By definition, multimedia streaming refers to a set of services with certain common characteristics. In short, a basic streaming system consists of sender and receiver applications that are interconnected via a packet-switched telecommunications network. The sender transmits a continuous flow of data packets containing compressed multimedia over the network to the receiver. Depending on the application, the receiver reproduces the multimedia data chunks immediately when they are available or after a short buffering delay.

To avoid starvation or buffer overflow, the encoded media data should be delivered to the decoder at the same rate the corresponding decoded data is consumed by the audio or video player. This is why the transport bit rate in a streaming system should be relatively stable and equal to the playback bit rate of the encoded multimedia data. How tight these requirements are depends on the application type: transport and buffering delays are not crucial for Internet radio or streaming of prerecorded audiovisual content at request (Video-on-Demand, VoD). On the other hand, interactive applications, such as Internet telephony and videoconferencing, set very strict requirements for the transport delay.

The quality requirements also depend highly on the application type. The subjective quality experienced by the end user in a multimedia streaming system depends mainly on the encoded multimedia bit rate and the data loss rate. Usually interactive applications allow reasonable quality degradation if only the carried message remains intelligible. In contrast, the quality requirements for entertainment audio and video may be very high.

Because of the wide variety of applications with different latency and quality requirements, it is sometimes difficult to make a clear distinction between downloading and streaming. A rough classification for multimedia delivery applications is outlined in Figure 1, based on interactivity (tolerance against delays) and perceived quality requirements. The primary focus of this thesis is on applications with relaxed delay requirements and relatively high quality requirements. This is the case with streaming, traditionally referring to non-interactive applications only, but there are also relevant use cases for teleconferencing with limited interaction and reasonable transport delay requirements. Therefore, the topics covered in this thesis could have some relevance also for teleconferencing applications.

Figure 1. Different multimedia content distribution applications classified by their quality and interactivity requirements.

2.2 Historical Perspective

2.2.1 From Traditional Data Communications to Real-Time Packet-Switched Networking

In the early years of digital communications, the roles of the traditional telephony and computer networking were clearly distinctive. In the traditional telephone system, the communicating parties must first establish a dedicated connection between each other before starting the conversation; this is called circuit switching. The communication is carried between the parties virtually in real time through a fixed communications circuit. In contrast, computer networking is based on a fundamentally different principle, namely packet switching. In the packet-switched networks, data is divided in packets and each packet is sent individually. Conceptually, the procedure is similar to the traditional postal service: data packets between computers are routed separately by specific devices, routers, according to the electronic address information included in each data packet.

Apparently, packet switching is optimal for carrying non-time-critical, occasionally appearing data bursts between computers, whereas circuit switching suits better for carrying steady flows of continuous data. However, the modern microprocessors with high processing power made demanding audio and video processing possible even in ordinary desktop PCs. The natural development lead to the idea of using computer networks for telephony and even broadcasting of audiovisual data to be raised. The first experiments for speech transmission in ARPANET, the predecessor of the Internet, have been reported in the late 1970s already [1].

IP protocol suite gained its position as de facto –standard in computer networking during the late 1970s and early 1980s. Gradually, it became apparent that the traditional data services would comprise only a small portion of the overall data traffic besides real-time multimedia transport in the IP networks of the future. RTP was proposed first in 1996 [2] and since that it has become the dominating protocol for real-time data transport in IP networks.

Growing demand for networked multimedia applications has also boosted the development of technologies supporting IP multicast and QoS differentiation. Multicast capability would be highly useful for TV or radio broadcasting over IP, but also for teleconferencing with a large number of participants. The basic protocols needed to extend the MAC layer multicast of Ethernet to work at the IP layer were developed in the late 1980s and early 1990s [3]. Since that, support for local multicast has been included also in several wireless systems. The fundamental problem of the multicast in wireless networks is the low reliability, due to the lack of link layer recovery of lost frames.

A lot of research has been carried out to scale the IP multicasting from local area networks to true multicast-enabled Internet. In 1992, the Internet's Multicast Backbone (MBone), comprising a subset of Internet routers with multicast capability, was created [3]. In mobile systems, multicast is more challenging issue, because the protocol should be able to deal with dynamic location in addition to dynamic group membership. This problem has been widely addressed in research lately and several different approaches and protocols have been proposed for mobile multicast [74]. Although there are promising architectures and implementations enabling multicasting in both wired and mobile environments, it is unlikely that large scale IP multicast will ever be universally available.

The network QoS is intended to get the benefits of both circuit and packet switching. The basic idea behind QoS is to provide priority service for the real-time applications by reserving network bandwidth for time critical data flows and forwarding priority packets first in routers, for example. The first proposals for QoS were based on resource reservation in routers via a dedicated protocol, Resource reSerVation Protocol (RSVP). This approach is called integrated services (IntServ), and it was first proposed in 1992 [4]. Latest research has concentrated mostly around a competing approach based on per packet prioritization, namely differentiated service (DiffServ) [5]. Several proposals have been made to support link layer QoS in various access technologies as well. At the time of writing this thesis, the practical QoS mechanisms are still evolving towards their full commercial applicability.

2.2.2 History of Digital Multimedia

Digital representation of audio and video is the basis for digital multimedia processing and communications. A large amount of data is needed to present digital audio and video in the uncompressed (raw) format, because the same number of bits is required for every individual time-domain audio sample and pixel of a video frame. For example, raw high quality stereo audio requires 16 bits per sample and 44100 samples per second for both channels (left and right), resulting in about 1.4 million bits per second. This is why the compression methods play an essential role in digital multimedia signal processing.

Two distinguished paths can be identified in the evolution of the audiovisual data compression: the codecs for digital transmission and the codecs for storage. The development of efficient speech coding technologies has been primarily driven by the rise of the digital circuit-switched telecommunications systems, whereas the generic audio and video coders have been designed mainly for storing the content on mass storage devices. Compact Disc (CD) was the first application making digital audio popular in the early 1980s. However, mass storage devices for computers were still rather expensive in those days. The raw digital PCM coding of the traditional CD could not fulfil the requirements for economical storage of audio in computer systems.

The perceptual audio coding paradigm proved its potential in the late 1980s and several proprietary codecs based on the paradigm were developed, including PAC by Lucent Technologies and AC-2 by Dolby Laboratories. Moving Pictures Experts Group (MPEG) was established in 1988 and since that it has played a major role in general audio coding standardization. MPEG-1 standard was published in 1992, including the Layer III general audio coding, known better as MP3. During the following few years, MP3 became extremely popular among the home users. MP3 provided near-CD stereo audio quality at bit rate of 128 kbps, which is significantly less than 1.41 Mbps of uncompressed audio. The substantially improved descendant of MP3, Advanced Audio Coding (AAC), was published as a part of the MPEG-2 standard in 1994. It provides almost the same quality as 128 kbps MP3 at the bit rate of 96 kbps. The latest version of AAC is included in the MPEG-4 standard and it is optimized even further and provides additional tools to improve the coding efficiency [6].

Although the MPEG audio coding standards have gained a dominating position in music compression, some audio coders are still challenging MP3 and AAC in digital music distribution. To name a couple of the most relevant rivals, Windows Media Audio (WMA) is a general audio codec developed and promoted by Microsoft and Ogg Vorbis is an open and free audio codec supported by the open source community [73].

In the digital radio systems, the available bit rate is typically low and the probability of bit errors is high. This is why the Pulse Code Modulation (PCM) coding used in wireline digital telephone networks cannot be used in cellular telephony as such. The design targets for speech codecs are low bit rate, reasonable quality and robustness against bit errors. Traditionally, the standardization sector of International Telecommunications Union (ITU-T, formerly CCITT) has been largely coordinating the standardization of speech codecs. However, the standardization efforts within European Telecommunications Standards Institute (ETSI) and 3G Partnership Project (3GPP) have increased along with the development of digital cellular telephony during the recent years. The first speech codec for GSM was 13 kbps Full-Rate (FR) standardized in 1989, followed by 5.6 kbps Half-Rate (HR) codec in 1995. The AMR codec, developed jointly by Ericsson, Nokia and Siemens, was adopted by 3GPP in 1999. Latest research advances in speech coding have enabled the use of wider audio bandwidth, which improves the audio quality significantly. The wideband version of AMR (AMR-WB) operates on several bit rates between 6.6 and 23.85 kbps [7].

One of the latest advances in audio coding is bandwidth extension based on spectral band replication. It makes coding of the high frequencies significantly more efficient and improves the audio coding performance especially at low bit rates [8]. AMR-WB+ and AAC+ are the enhanced versions of the original AMR-WB and AAC codecs, using bandwidth extension. They represent the state-of-the-art in speech and audio coding at the time of writing.

The history of digital video compression is not this long because of the prolonged prevalence of the analogous television and VHS. First implementations for digital video processing were purely proprietary. Standardization activities for digital video coding were started as late as in the 1980s, resulting in the CCITT recommendations H.120 and H.261. Since MPEG was established, it has taken an active role in the standardization efforts related to digital video coding together with ITU-T. In practice, ITU-T and MPEG have developed video coding technologies in a joint partnership and the relevant ITU-T standards for video compression are technically identical to their MPEG counterparts [9, 10, 11]. In this thesis we focus on the MPEG standards.

MPEG-1 video coding targeted for digital storage media at bit rates up to 1.5 Mbps. The coding improvements present in MPEG-2 provided better support for visual communications applications, such as digital cable television [9, 10]. The latest enhancements in video coding have improved the compression ratio significantly, which enables real-time delivery of video content even over wireless links of low bandwidth. Many of these advanced features have been adopted by the MPEG-4 Advanced Video Coding (AVC) standard. AVC includes different coding and network adaptation tools for wide range of applications from content storage to conversational video telephony. AVC supports video bit rates from 64 kbps, suitable for low rate visual communications, up to 240 Mbps of very high quality video [11].

2.3 Real-Time Transport Protocol and its Companions

2.3.1 RTP Framework

A basic packet-switched network service lacks several functionalities required from a real-time data delivery mechanism, such as timestamps for relating the content in a packet to the intended playout time, and sequence numbers for detecting packet losses and rearranging packets arriving in wrong order. Real-time Transport Protocol (RTP) is the strongly dominating protocol for carrying data with real-time nature over packet-switched networks. It defines the mechanisms needed for basic real-time communications, including synchronization, packet reordering and source identification.

The latest version of RTP has been published as IETF RFC 3550 [12], containing only slight revisions to the obsolete IETF RFC 1889 [2]. It is usually, but not necessarily, located above UDP in the IP protocol stack. Basically, RTP can be used with any kind of real-time application, but has been designed especially with multicast teleconferencing applications in mind. Real-time Transport Control Protocol (RTCP) is co-operating with RTP to convey statistical information about connection between the communicating parties. Figure 2. shows a typical protocol configuration for real-time applications using an IP network.

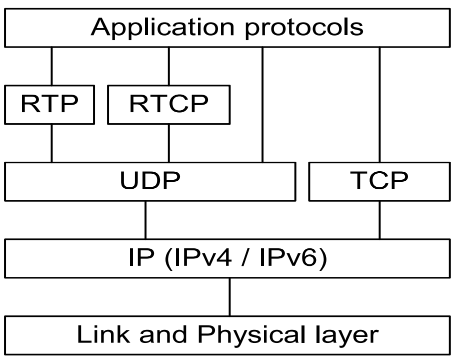


Figure 2. Typical protocol configuration for real-time transport over IP networks.

Figure 3. illustrates a typical RTP usage scenario. The end systems generate and consume the real-time content. There may be also mixers and translators involved. Mixers receive RTP packets from different sources, combine them in some manner, potentially change the encoding, and forward the new RTP packets. This might be useful in teleconferencing, for instance. Translators may modify the

real-time content somehow, such as convert encoding, without mixing. This facilitates transmission in a heterogeneous network environment, because different encoding and bit rate can be used for streaming the same content over different types of network. In practice, the role of the mixers and translators in RTP communications has remained small.

RTCP works in conjunction with RTP to convey feedback information between the communicating parties within RTP session. Every sending node transmits occasionally RTCP Sender Report (SR) messages to inform the other nodes about the number of RTP packets it has transmitted. Correspondingly, receiver nodes transmit RTCP Receiver Reports (RR) to inform the others about the receiver statistics, primarily the RTP packet loss rate and the variation in relative transport times (jitter). RTCP allows every sender to adapt their mode of operation to the prevailing conditions. For example, if the packet loss rate indicates congestion, streaming server can switch to a lower bit rate encoding mode. It is also possible to define application-specific RTCP messages. Therefore, RTCP can be used to carry also non-standard control commands, such as retransmission requests.

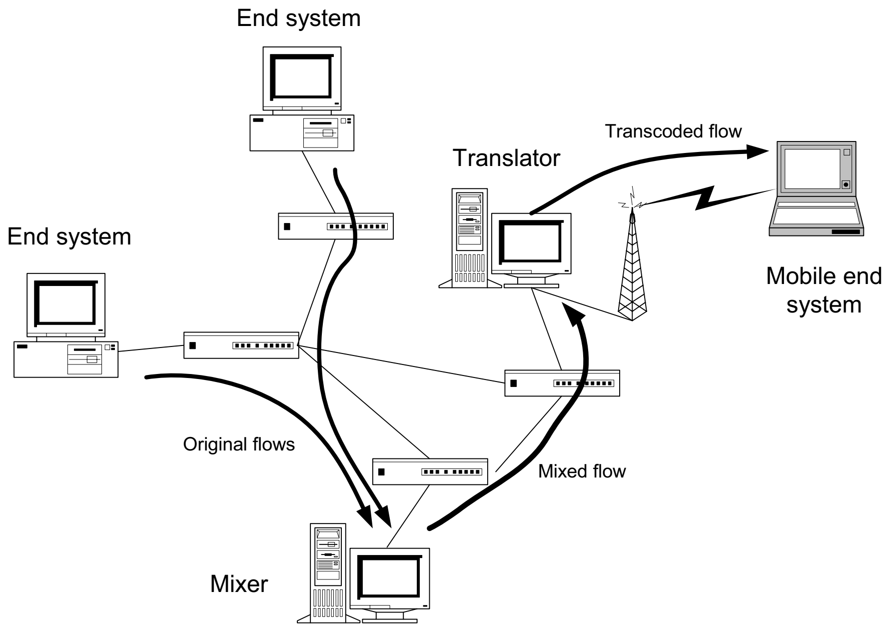


Figure 3. Example RTP usage scenario.

RTP specifications leave many implementation details open. This is why separate RTP profile definition and payload format specification is needed in addition to the RTP specification when the protocol is implemented for a certain audio or video codec. The additional documents specify details such as rules for generating the RTP payload out of the encoded data, timestamp resolution and application specific extensions to RTP.

2.3.2 RTP Extensions

Several extensions have been proposed to optimize the performance of RTP under difficult network conditions. Because RTP as such does not guarantee reliably delivery of data, RTP extensions usually aim to facilitate recovery in case of packet losses. Generally speaking, there are two methods for increasing reliability: retransmissions and Forward Error Correction (FEC). There are proposals based on both approaches to improve RTP.

The simplest way to implement FEC is to define a payload format that allows same data to be transmitted multiple times. There are also more efficient FEC schemes. In RFC 2733 [13] a generic FEC scheme has been defined. It uses the binary exclusive or (XOR) operation to generate FEC packets out of two regular RTP packets. If one of the RTP packets is lost, the data can be reconstructed by performing the XOR operation for the received RTP packet and the FEC packet.

An example of a payload format for media specific FEC is defined in RFC 2198 [14]. This payload format allows a redundant secondary frame coded at a lower accuracy than the primary frame to be transported in a different RTP packet. Because the secondary frame is much smaller than the primary frame, redundancy overhead can be efficiently reduced. If one RTP packet is lost, the associated frame can still be reproduced at lower quality using the secondary frame that is (hopefully) received correctly.

The price to pay for the improved reliability when using FEC is the increased network overhead, which may lead to unwanted network link overload and congestion. Depending on the codec, all data in an RTP payload is not always equally important. It might be sufficient to protect only the most critical data sections in each RTP packet with FEC. In this case, the redundancy overhead can be kept significantly smaller than in the generic FEC. This kind of mechanisms are referred as Unequal Error Protection (UEP).

FEC cannot guarantee full reliability: it is always possible that the redundant backup data gets also lost. More efficient network resource utilization and better loss recovery rate can be achieved with retransmissions. However, the use of retransmissions is problematic with real-time applications because of retransmission delay. In addition, simple end-to-end retransmission schemes cannot be used with multicast applications as such, because feedback messages and retransmissions in a large multicast group may cause very significant network overhead. The problems related to reliable multicast have been addressed in [15].

There are, however, a number of scenarios where limited use of retransmissions can be highly beneficial even in real-time communications. These include applications with relaxed latency requirements, such as unicast audio/video-on-demand, or even teleconferencing and Internet broadcasting within a small multicast group. This is why there is a proposal to extend RTP with retransmissions [16]. The scheme is based on the selective retransmission paradigm, which allows the server to retransmit the most critical RTP packets in case of heavy packet loss ratio. The selective retransmissions can easily be tailored to suit the requirements for different applications.

2.3.3 Related Protocols

RTP is a transport protocol. It does not provide means for exchanging control commands and session information, such as codec parameters. However, in the IETF streaming framework there are other protocols for these purposes. Real-Time Streaming Protocol (RTSP) [17] has been designed for sending control commands, for example to start and stop streaming and set up a session. RTSP can also be used to convey codec dependent information, such as the audio sampling rate or coding profile, encapsulated in Session Description Protocol (SDP) messages [18].

Although RTP does not contain any specific requirements about the lower layer capabilities, use of reliable connection-based protocols is generally considered inappropriate for real-time transport. The main reason is that the connection-based protocols, such as TCP, use retransmissions to recover from packet losses and congestion control slowing down the transmission rate when packet losses occur. Because of these mechanisms, the predefined natural transmission rate required by real-time applications cannot be guaranteed. Another drawback of TCP is that it cannot support multicast. Because of these reasons, UDP is usually employed to carry RTP traffic in IP networks.

However, even a connectionless transport protocol cannot guarantee timely and reliable delivery of packets as such. Congestion in the network can cause packet losses and increased transport delay. To overcome these problems, suitable QoS mechanisms could be employed to prioritize RTP traffic in the network. In wireless systems, mobile terminal position is often dynamic. This is why the connection-oriented QoS mechanisms based on end-to-end resource reservation do not seem to suit well for the mobile IP networking. Nevertheless, several extensions to QoS mechanisms and signaling protocols based on RSVP have been proposed to provide better support for mobility [75].

UDP uses a 16-bit checksum to guarantee the integrity of the datagram content. By default, all UDP datagrams with bit errors are discarded. However, many multimedia applications could benefit from getting damaged data instead of losing the whole datagram. UDP checksum can be turned off, but this is highly discouraged, because in this case also the UDP header may corrupt, leading to unexpected behavior on the transport layer. For this reason, UDP Lite [19] has been proposed. It allows partial checksumming, which makes it possible to protect the most vulnerable part of a datagram and leave the error resilient part unprotected [20, 21]. Typically, the protected part would include the protocol headers for UDP and RTP as well as the codec-specific header in the RTP payload.

2.4 Digital Coding of Multimedia

2.4.1 Basic Principles

There are a number of different audio and video coding paradigms. First of all, the fundamental division can be made between lossless and lossy compression methods. When lossless coding is used, a bit-exact replica of the original data can be reproduced in the decoding process. Lossy coding methods do not even aim to preserve all the details of the original content; however, the subjective difference between the original content and the encoded content after decoding is intended to be as small as possible.

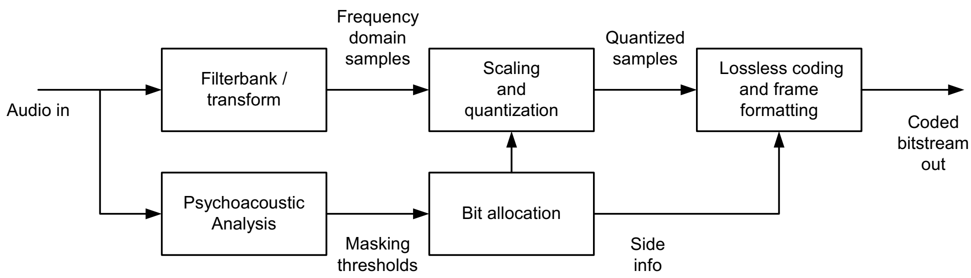
The most advanced multimedia coding standards use both lossless and lossy compression techniques in parallel. In the lossy coding phase the components of audio or video are quantized according to their perceptual relevance. In this way, a different number of bits can be allocated for coding different parts of the data, depending on the importance. In the last phase, the quantized components are compressed using lossless coding methods.

The most appropriate lossless compression method is Variable-Length Coding (VLC), more specifically the Huffman coding. It is based on the typical proportional prevalence of each different symbol to be encoded: some symbols are supposed to appear more often than others. Each symbol is turned into a Huffman codeword. The most common Huffman codewords contain the smallest number of bits.

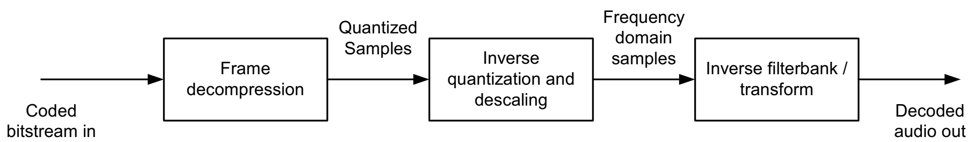
The most serious drawback in VLC is the vulnerability to bit errors. Because the number of bits allocated for each symbol is not known *a priori* during the decoding process, the beginning position of each codeword is not known before the previous codeword has been decoded. This is why a single bit error may be fatal if the length of the mutated codeword is different from the length of the original codeword.

2.4.2 MPEG Audio Coding

The dominating paradigm for general audio coding today is perceptual audio coding. It is based on the idea of eliminating the frequency components that cannot be perceived by a human ear. Because no bits are used to store the perceptually irrelevant frequencies, high coding efficiency can be gained. For example, MPEG Layer III (MP3), MPEG AAC, WMA and OggVorbis are all perceptual audio codecs. The generic structures of a perceptual audio encoder and decoder are sketched in Figure 4. A transform block of a perceptual encoder generates a frequency domain presentation of the initial PCM audio signal. A psychoacoustic analysis is performed to define optimal bit allocation for each frequency component. The frequency domain samples are scaled and quantized before lossless coding and frame formatting.



a) Generic perceptual audio encoder.



b) Generic perceptual audio decoder.

Figure 4. Basic structure of the generic perceptual audio encoder and decoder.

The psychoacoustic analysis is based on the experimentally observed characteristics of the human auditory system. First of all, a loud signal makes lower sound impossible to hear if the frequencies of the two signals are close to each other. This is called masking: a loud signal masks other signals. The masking effect applies in both frequency and temporal domain and it is one of the most important phenomena in the perceptual audio coding: it is not reasonable to use bits for encoding the frequency components below the masking threshold. A perceptual model is used to compute the masking thresholds in different cases. Frequency domain masking is depicted in Figure 5.

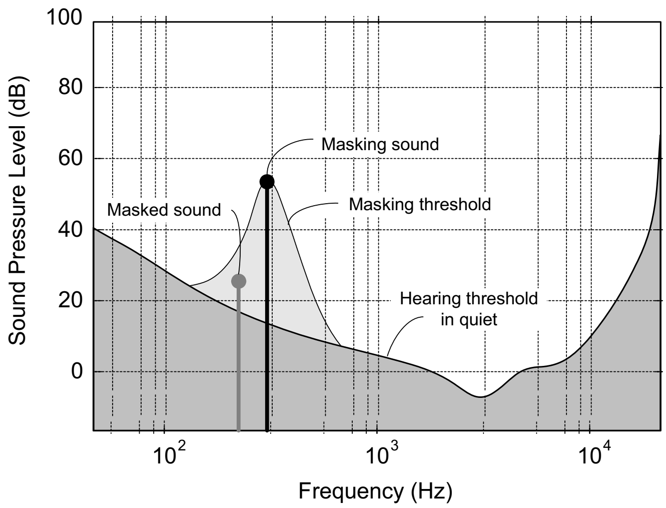


Figure 5. Masking effect illustrated in frequency domain.

In typical perceptual audio codecs, such as MPEG general audio coding, the audible frequency band is divided in subbands with different scaling for the actual frequency components. The range for the frequencies in each subband is defined by scalefactors. MP3 format defines a static configuration for the subband division: there are 32 subbands with 36 frequency samples per each, resulting in 1152 spectral coefficients per frame. The quantized and scaled spectral coefficients are Huffman coded.

MPEG-2 AAC is based on MP3, including numerous improvements. Modified Discrete Cosine Transform (MDCT) is used for transform from the time to frequency domain, just like in MP3. AAC provides dynamic scalefactor bands with different lengths and Huffman codebook indices. Each scalefactor is coded using DPCM coding first and Huffman coding to the DPCM values. The quantized and scaled MDCT coefficients are Huffman coded so that each Huffman codeword represents two or four adjacent coefficients, depending on the applied Huffman codebook.

MPEG-4 AAC provides several optional tools improving the MPEG-2 AAC. The Temporal Noise Shaping (TNS) tool controls the fine structure of the quantization noise within each filterbank window, helping to avoid perceptually annoying pre-echo in transition periods. The Perceptual Noise Substitution (PNS) tool allows very efficient coding of the noise-like signal components. The Long-Term Prediction (LTP) tool intends to eliminate the redundancies between adjacent frames. It is useful especially when the audio contains stationary harmonic tones.

MPEG-4 introduces also optional Error Resilience (ER) tools improving the robustness against bit errors. ER tools allow protection of the most vulnerable critical bits via efficient FEC. DPCM values of the scalefactors are coded using Reversible Variable-Length Codes (RVLC) instead of the traditional Huffman codes [22]. The RVLC codewords are symmetric, and therefore they can be read from the end to beginning. This approach allows the decoder to resynchronize decoding process if bit errors are detected. The coded spectral coefficients are protected against error propagation with a Huffman Code Reordering (HCR) tool. This tool allocates the most significant priority codewords in predefined positions. The gaps left between the priority codewords are filled with the remaining non-priority codewords. This method effectivetly restricts the error propagation in the Huffman coded data sections [23].

2.4.3 MPEG Video Coding

There is typically a significant amount of statistical and subjective redundancy between consecutive video frames. This is why prediction mechanisms play a major role in the modern video coding standards. For encoding purposes, a video frame is divided in blocks. The basic unit in encoding is a block of 8x8 pixels. An appropriate transform, such as Discrete Cosine Transform (DCT), is applied to each block in both horizontal and vertical direction to remove the spatial redundancies within each block. After the transformation, data can be encoded efficiently.

The modern video coding standards adopted in the MPEG standards divide video frames in three classes: I-, P- and B-frames. I-pictures contain all the information needed to reproduce the video frame. In contrast, P- and B-frames are predicted from the neighbouring frames. An efficient method for prediction is motion compensation. Moving objects are extracted from the background and the difference in object position between frames is coded as a motion vector [10, 24].

MPEG-4 AVC comprises a number of different tools optimising the coding performance in various usage scenarios. Compared against the MPEG-2 video, the motion compensation and prediction mechanisms have been significantly improved. In addition, the standard allows adaptive use of different transformation block sizes and two different options for the entropy coding. There are three different profiles defined in the standard, each including a different set of features [11].

Due to the advanced features and complicated bitstream structure, advanced video coding is vulnerable to bit errors and data loss. In MPEG-4 AVC a special attention has been paid for the robustness against data errors and losses. Synchronization markers and robust parameter set structure facilitate the bit error recovery. Network Abstraction Layer (NAL) is an essential part of MPEG-4 AVC, which is missing in its counterpart standard for audio coding. NAL is designed to provide useful features for networked video. The standard allows each video frame to be divided in independent slices of flexible size, enabling these slices to be transported and even interleaved separately from each other in different NAL units. Robust networking is supported in the standard also via flexible macroblock ordering, sending redundant regions of pictures and improved synchronization [11].

2.5 Future Directions and Challenges in Multimedia Streaming

Multimedia streaming is still a hot research topic both in the academia and industry. The ongoing research activities can be classified roughly in coding oriented and network oriented tracks. The current important issues in multimedia coding include especially scalability in different forms. The related issues in networking comprise network support for QoS, proxy-assisted streaming and traffic engineering. In the wireless domain, energy efficiency is a significant emerging topic. In this Section, the ongoing activities and trends in the multimedia streaming are studied.

2.5.1 Topics in Multimedia Coding

Scalability is one of the most relevant issues in the multimedia streaming nowadays. Scalable coding is useful in applications that need to adjust the transmission rate according to the network conditions. In the traditional wireline IP networking, packet losses indicate congestion. This is why the well-behaving TCP-friendly applications should react packet losses by decreasing the transmission rate. The most straightforward method to implement an adaptive transmission scheme at the server is to use several redundant content files encoded at different bit rates. The server can then switch between different files to adapt with the changing network conditions.

However, Fine Grain Scalability (FGS) provides much more sophisticated solution for the bit rate adaptation. The scalable codecs allow data to be removed in the end of each frame. The frame would still be decodable, although the bit rate and quality were lower. Scalable coding can also facilitate transcoding. In a typical scenario, a streaming proxy receives a multimedia stream at a high bit rate and forwards it to a narrowband access network, such as a cellular radio network. Without scalable coding, the proxy should first decode the stream and then re-encode it using different coding parameters or even a different coding standard to reduce the bit rate. When scalable coding is used, the proxy could just remove the appropriate part of each frame.

Scalable video coding has been studied extensively from the end of 1980s, and MPEG-4 AVC supports fine-grain scalability already [24]. Less effort has taken place to bring scalable audio coding in standards. Several proprietary coding methods have been proposed to achieve fine grain scalability in audio coding [25-27], but the performance of the scalable audio coding supported in MPEG-4 is still relatively weak especially at the lower bit rates [28].

Layered coding is another important subclass of scalable coding. When the layered coding paradigm is applied, encoded multimedia frames consist of a base layer and one or more enhancement layers. The base layer alone is sufficient to reproduce the frame at the minimum quality. The enhancement layers can be used to improve the quality. Layered coding is useful especially for multicast applications: it is possible to transmit the base layer stream only to the receivers behind a narrowband access links. When the multicast group for the enhancement layer transmission is a subset of the multicast group for the base layer transmission, users of a broadband access networks can enjoy the full quality as they receive also the enhancement layers. In this scenario, the server can avoid transmitting two different, but partially redundant streams. Layered coding facilitates also packet loss recovery, because the loss of the enhancement layer data does not cause gap in reproduction, but only decreases the quality temporarily.

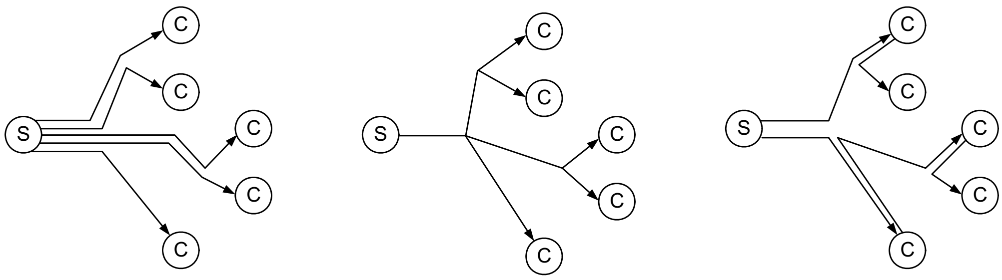
In Multiple Description Coding (MDC), the encoder produces several representations (descriptions) of each frame. Unlike in layered coding, the different descriptions are equal in importance. Each description is sufficient to reproduce the original frame. However, the quality is better when there are several descriptions available. This approach is useful when the communication link suffers from uncorrelated packet losses. When different descriptions are allocated in different packets, it is unlikely that all packets containing a description of a certain frame get lost. A thorough survey of MDC is available in [29]. Although the concept of scalable coding is no means new, it is still a challenging research topic due to the continually advancing multimedia coding and networking technologies.

2.5.2 Topics in Multimedia Networking

The network QoS has been an extensively researched topic in networking during the last few years. Traditionally, the aim of the network QoS is to provide differentiated service to different applications, according to their needs. Real-time interactive applications should have a privilege over the conventional data applications, because the interactive applications are much more vulnerable to packet delays. Several schemes have been developed to provide service differentiation on different layers, from physical access links up to the application layer [30].

The next challenge in the network QoS is to bring the theory in practice. There are several open questions, regarding signalling, billing and cross-layer mapping of QoS parameters, and these issues are rather political than technical. It is difficult to implement fair policies to provide end-to-end quality differentiation. A simple method is to allow users to reserve resources with RSVP or classify all RTP packets as priority packets. This would lead to temptation to oversubscribe resources [31] or implement proprietary fast browsing or peer-to-peer file sharing applications using RTP for non-real time data transport, ruining the idea of the service differentiation. Another approach is to provide better service for those who pay extra for the improved QoS. The Internet is an open network, consisting of several subnetworks and trunks administrated by different companies and institutions. Even though several end-to-end QoS architectures have been proposed [32, 33], there are serious considerations for the practical implementation of these schemes because of the lacking or inadequate support for incremental deployment and coordination between different network architectures [34].

Because of these problems, many are sceptical about the universal network QoS provisions in the Internet. However, different QoS schemes may still play an important role in specified environments, such as wireless access networks and ad hoc networks. For example, in cellular radio networks the radio access link between the base station and the terminal is almost always the bottleneck. In this case, it is reasonable for the telecom to provide its customers with different levels of QoS even if the Internet in the background is still based on the best effort service. This is why QoS is still one of the most significant research topics in multimedia networking.



a) Unicast b) Multicast c) Application layer multicast

Figure 6. Unicast, multicast and application layer multicast compared. S denotes the originating server (data source) and C denotes a client.

Peer-to-peer streaming and application layer multicast is an important new research area regarding multicast streaming applications, such as TV broadcasting or multiparty teleconferencing [35]. Many commercial stakeholders have been disappointed about the slow deployment of multicast network architectures, such as MBone. Peer-to-peer streaming is based on the idea that each receiver may act as a forwarding proxy. This is how the original data source does not need to send unicast data streams to all receivers directly, but the burden can be shared between the peers. As a result, the network load can be reduced just as in traditional multicast, but without support for the multicast in the network layer. Figure 6 compares unicast, traditional multicast and application layer multicast.

Application layer multicast enables several interesting schemes. For example, MDC can be used to produce several descriptions of the source data that is distributed via different routes. This kind of technique allows an efficient combination of traffic engineering and packet loss resilience, especially when different links suffer from different error rates. Several alternative approaches have been proposed to implement peer-to-peer streaming, but there are still open research problems regarding the application layer multicast. For example, it is challenging to find the optimal distribution of the traffic burden when the bandwidth resources are heterogeneous and the receivers have different intentions for donating bandwidth for the other users [36].

2.5.3 Emerging Topics: Moving to Wireless Domain

As the wireless telecommunications evolve, multimedia streaming over wireless IP networks is getting more and more attention. The topics discussed above are mostly independent on the carrier medium. However, there are also several interesting new issues in multimedia communications that are specific to the wireless domain.

Power efficiency is one of the critical factors in mobile computing, because the battery technology is evolving much slower than the available memory and CPU speed [37]. To increase the battery lifetime for the mobile multimedia applications, both energy efficient radio communications and multimedia processing are needed. Nowadays, the modern processors provide capability of adjustable voltage and frequency levels. The user can select a lower speed and lower power consumption when there are no time critical tasks being processed [38].

Power-aware radio communications is another emerging area in the power efficiency research. Typically, the radio interface of a mobile receiver consumes significantly more power in the active state than in the sleep state. Unfortunately, the radio interface is not able to receive data in the sleep state. In the modern WLAN standards, this problem has been solved by deploying a power save mode, where the receiver wakes up periodically to probe whether there is a station that wants to transmit data to it [39].

However, this solution works only for occasional traffic bursts with relatively long intervals between packets. Long periods in the sleep mode between packets are usually not allowed in a constant bit rate streaming system, because the packet interval is short and there is not sufficient time for the radio interface to switch between the sleep and active states. There are proposals to solve the problem by using a local proxy that reshapes the traffic so that the packets are transmitted over the radio link in bursts [40-45]. This kind of transmission mode allows longer periods of sleep between the bursts. The topic is still under active research.

Another relevant issue in wireless domain is bit error management. In a wireless medium, bit errors are much more common than in wireline networks. However, the erroneous packets are usually discarded by the error checking mechanisms either at the link layer or at the UDP layer at latest. In brief, the losses in the traditional fixed IP networks are usually related to congestion, whereas losses in the wireless links are more likely to be caused by physical transmission errors.

In traditional data communications, the user expects to receive data without any errors. However, in real-time communications, users may have different preferences. Many video and audio codecs can cope with a reasonable number of bit errors in the content. In this case, it can be better to deliver the erroneous RTP packets up to the application layer rather than discard the whole packet. Usually, there are link layer retransmissions used to recover damaged packets. Even in this case, it could be useful to disable the bit error detection, because the link layer retransmissions may decrease the overall capacity of the shared radio link. UDP Lite [19] was proposed to mitigate the problem.

The idea of using the bit error characteristics of a radio channel has also reflections to the congestion control and packet loss differentiation. If the bit error detection is switched off at the lower layers, the application may use its own checksums to decide whether the network problems are mainly related to bit errors or congestion [46, 47, P5]. This information can be useful to choose between different streaming options and strategies. If packet losses are not appearing because of congestion, it may be a reasonable strategy to use FEC or retransmissions instead of reducing the transmission rate.

Chapter 3 Application Layer Optimization of Audio Streaming

Even if a streaming application cannot control the network QoS, different application layer schemes can be used to support real-time transport and facilitate error recovery in the case of packet loss. In this Chapter, these techniques are studied. Especially, the advanced scheme for streaming perceptually coded high-quality audio proposed as a part of the thesis is addressed. In short, the scheme is based on the diversity of the internal components in each AAC audio frame. By interleaving or shuffling these data components among different RTP packets, the robustness against packet loss can be significantly improved. The dedicated transport and error concealment strategies for this kind of system are also explained.

3.1 Error Recovery and Concealment Strategies

Different application layer strategies to recover from packet losses in an audio streaming system can be categorized in receiver-based and sender-based error management techniques [48]. The receiver-based techniques comprise the error concealment strategies that do not require any actions from the sender. These techniques are often referred as error concealment, because they do not intend to recover the original data that is missing, but solely mitigate the perceived quality degradation caused by data loss via signal processing means. In contrast, the sender-based techniques rely on assistance by both the sender and receiver, including FEC and selective retransmissions. Receiver- and sender-based techniques are often used to complement each other to achieve optimal performance.

3.1.1 Receiver-based Error Concealment

Purely receiver-based error concealment strategies can typically be used if the expected packet loss rate is low and the requirements for audio quality are not overwhelming. The traditional techniques in this category include muting and frame repetition. Muting is the most simple error management strategy, where the missing audio frames are just replaced with silence. Substantially better results have been achieved with the frame repetition method that uses the previous correctly received audio frame as a replacement for the lost frame.

More sophisticated version of the simple frame repetition is the content-based frame replacement. For example, if a missing audio frame expectably contains a drumbeat, it may be a good strategy to use the previous drumbeat as replacement [49]. Another advanced error concealment strategy is interpolation. Missing audio sequence can be predicted using mathematical interpolation from the audio data before and after the missing clip [50, 51]. Interpolation can be implemented in either the time domain [50] or the frequency domain [51]. The weakness of interpolation, especially in time domain, is the complexity, both in terms of the implementation effort and required processing power.

3.1.2 Retransmission-Based Error Recovery

Only protocols employing retransmissions can provide fully reliable recovery of the missing data. In addition, the redundancy overhead, that is characteristic to FEC, can be avoided by using retransmissions. However, it is not always feasible to use retransmissions. First of all, a feedback channel is needed to convey the negative or positive acknowledgements (NAKs or ACKs), but it is sometimes missing from a broadcast type of transmission medium, such as digital cable TV. In IP broadcasting and multicasting, the use of retransmissions is limited as well, because multiple retransmission requests and retransmissions could easily cause overwhelming traffic in the network. This problem is called feedback implosion [15].

In highly interactive applications, retransmissions should also be avoided altogether. If the end-to-end transport delay is long, the application cannot afford any more delays caused by retransmissions. This is an issue especially in Internet telephony and teleconferencing. In this dissertation, we focus primarily on the streaming applications with more relaxed latency requirements. Therefore, retransmissions can be considered as an option.

There are several different retransmission-based transport protocols. A fully reliable transport mode is usually not needed for a streaming application. This is why streaming applications rely often on NAK-based retransmission mode with a limited number of retransmission attempts. One possibility is to use the selective RTP retransmissions relying on RTCP-based feedback mechanism [16]. In this kind of system, the application can decide to retransmit only the most critical packets. Traditionally, the concept of selective retransmissions is especially useful in video streaming, because a video stream typically consists of frames with different priorities.

3.1.3 FEC-Based Error Recovery

FEC is a generic term describing error correction mechanisms that rely on added redundancy. There are several methods to implement FEC, both media independent (generic) and media specific [52]. Generic FEC is intended to protect the RTP payload data regardless on the media type. Media specific FEC uses different media formats for the primary and secondary (redundant) media stream. The two frames, primary and redundant, are allocated in different RTP packets and the redundant frame is used only if the original frame gets lost. The playback quality of the redundant stream does not need to be as high as that of the primary stream. This allows the system to reduce the redundancy overhead, because the size of redundant media frames can be smaller than the size of the primary frames.

The most straightforward generic FEC scheme is to duplicate each data packet. In this case, the data is lost only if every redundant packet is lost. In this kind of scheme, the redundancy overhead may easily become a problem. More complex linear error correcting codes can be used to achieve lower level of protection at smaller overhead. The XOR-based parity coding and Reed-Solomon coding are examples of this kind of codes.

In a simple parity-based FEC, the binary eXclusive-OR (XOR) operation is applied over a sequence of data units and the result is used as redundant FEC data. In this scheme, any single loss of a data unit can be reconstructed by applying XOR operation to the remaining units plus the FEC unit [13]. There is also a more recent proposal for generic FEC supporting unequal level protection [53]. This is a useful feature in case the original RTP payloads consist of data sections with different priorities.

More advanced codes, such as Reed-Solomon, allow more flexible configurations of the data units and corresponding FEC units. This kind of codes are linear codes, consisting of *k* data units and *n* corresponding FEC units, respectively. These codes are capable of recovering from any loss of *n* units out of (*k*+*n*) units in total [54].

3.2 Interleaving and Shuffling of Data Elements

3.2.1 Background

Interleaving is a well-known method for improving the error resilience. According to the common knowledge, long gaps in audio playback are perceptually more disturbing and more difficult to conceal than short gaps spread over a longer time frame. This is the rationale behind the use of interleaving and its different variations in telecommunications.

Simple interleaving schemes for packet audio have been developed in the early 1980s already. Jayant et. al. proposed a system that allocates odd- and even-numbered voice samples in different packets. If one of the packets is lost, every second sample is lost, but they are easy to recover with the interpolation [55]. Another similar concept with a more sophisticated sample recovery method based on pattern matching has been proposed by Yuito et. al. [56].

The early proposals for the use of interleaving with packet audio are based on waveform audio codecs, such as PCM or DPCM, and thus they operate purely in the time domain. In this thesis, the primary focus is on the streaming of perceptually coded audio. The advanced audio compression standards, such as MP3 and AAC, operate in the frequency domain. Figure 7 illustrates the general idea of shuffling the data elements from each frame among several packets. When a perceptual audio codec is used, each data element typically represents a frequency component. One packet loss makes one component to be lost in several adjacent frames, which causes perceptually less severe distortion that is significantly easier to conceal than loss of an entire frame.

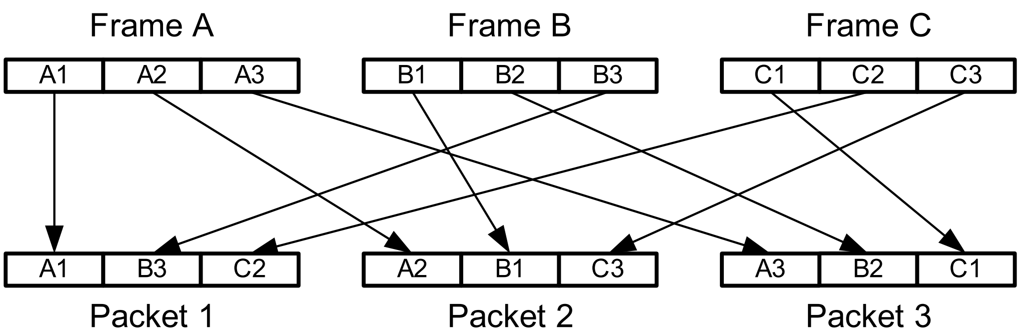


Figure 7. Basic idea of interframe data shuffling or interleaving.

Unfortunately, the bitstream formats forf the advanced coding standards are typically very complex. Each frame contains several elements with a different purpose and level of significance. For these reasons, it is not as straightforward as shown in Figure 7 to develop an interleaving scheme for streaming perceptually coded audio.

In perceptual audio coding, the different data elements in an audio frame can typically be classified roughly in three categories: critical data, scalefactors and spectral coefficients. The critical data includes the essential side information for the decoding process, such as Huffman codebook indices, transform window type and different flags. Without this data, the frame cannot be decoded. Scalefactors define the scaling for the different frequency bands. Spectral coefficients include the quantized frequency samples. This classification is the basis for designing an interleaving scheme, as each data category should be handled differently.

3.2.2 AAC Data Shuffling

Figure 8 shows the frame structure of an AAC stereo audio frame. In the sake of clarity, the illustration is highly simplified; in practice, there would be many more elements. The header data is allocated in the beginning of each frame. After the header, there are two chunks of channel specific data, for the left and right channel, respectively. Both chunks contain critical data, scalefactors and spectral coefficients. The critical data define scalefactor band allocations and the Huffman codebooks used for each scalefactor band. The scalefactors define the scaling within each band. The spectral coefficients contain the Huffman coded and quantized MDCT coefficients.

Limits for each scalefactor band, as well as the number of the quantized scalefactors, are defined by the critical data. This is why the scalefactors and spectral coefficients cannot be decoded without rendering the critical data first. A lost or damaged critical section makes the whole frame useless. On the other hand, each scalefactor is required for de-quantizing the spectral data. The decoder cannot de-quantize spectral coefficients accurately, if the associated scalefactor is missing or erroneous. Failing in de-quantization can lead to very annoying perceptual artifacts. The dashed arrows in Figure 8 show the dependencies between the scalefactors and the respective Huffman coded spectral samples.

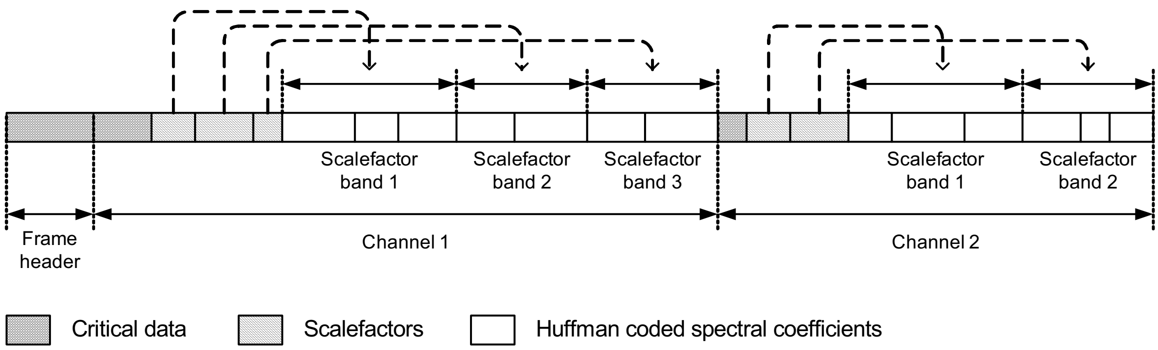


Figure 8. Simplified AAC frame structure.

For robust audio streaming, it is possible to extract the individual elements and shuffle them among several packets. However, because the proportional importance of the elements is different, the implementation details for the interleaving scheme must be considered carefully. Different level of protection against the data loss should be allocated for different classes of data. Apparently, the critical data requires the best protection. Basically, both FEC and retransmissions can be used to implement the uneven protection of data sections in the transport mechanism. Different alternatives for implementing these mechanisms are discussed more detailed in Section 3.4.

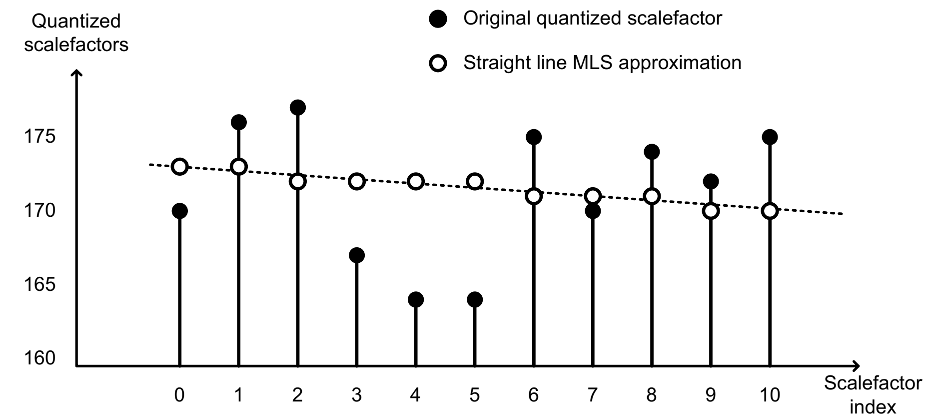
3.3 Optimization of AAC Codec

The baseline AAC coding standard is not optimal for the transport scheme proposed in our work. This is why the AAC bitstream format has been modified to suit better for the proposed interleaving and transport schemes. The major modifications comprise an improved scalefactor coding scheme, interleaving of MDCT coefficients among Huffman codewords and allocation of Huffman codewords in slots of a predefined size.

3.3.1 Coding of Scalefactors

The scalefactors in AAC are coded in three phases. First, a scalefactor is quantized, resulting in an integer between 0 and 255. Second, DPCM is applied to the quantized scalefactors. Only the first scalefactor in each frame is stored as such; all the rest values indicate the difference between the previous scalefactor and the current one. In the third phase, the redundancies are removed by coding the DPCM values with Huffman coding.

This kind of coding scheme is extremely vulnerable against errors. Huffman coding is prone to bit error propagation. DPCM coding is prone to data losses as well: if one DPCM code is missing, all the quantized scalefactors following the missing one would be incorrect, unless the lost DPCM code was zero. This is our motivation for using a different kind of scalefactor coding.



Original scalefactor	170	176	177	167	164	164	175	170	174	172	175	...
DPCM value	170	+6	+1	-10	-3	0	+11	-5	+4	-2	+3	...
Straight line approximation	173	173	172	172	172	172	171	171	171	170	170	...
Prediction residual	-3	+3	+5	-5	-8	-8	+4	-1	+3	+2	+5	...

Figure 9. An example of using the original and proposed scalefactor coding methods in AAC. The first DPCM value gives the first original quantized scalefactor (i.e. "global gain").

In the proposed scalefactor coding scheme, the quantized scalefactors are first approximated roughly by a straight line. The line is fitted to the quantized scalefactors using Minimum Least Squares (MLS) method. The offset and angle of the curve are coded with 13 bits altogether. In the next phase, the prediction residuals (the difference between the MLS approximation and the actual quantized scalefactor) are Huffman coded. The original and proposed scalefactor coding schemes are illustrated in Figure 9. More detailed description of the scheme is given in [P3].

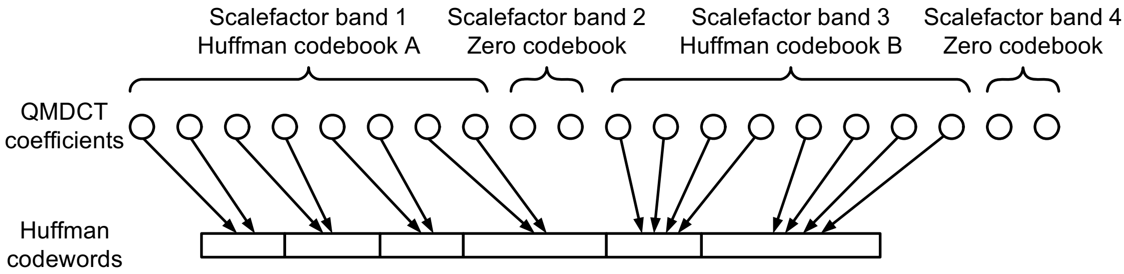
The proposed scheme decreased the coding efficiency only marginally when the scheme was applied to four different test music samples. Even in the worst case, the average frame size increased by only 1.7%. In comparison to the conventional scalefactor coding, the error resilience was substantially improved [P3].

The lost scalefactors can be replaced rather straightforwardly. Due to the characteristics of the MLS method, the sum of the prediction residuals is zero (or at least close to zero, depending on the quantization errors). If there is only one scalefactor missing, it can be reconstructed by using the value that makes the sum of the residuals to zero. In case there are more than one residuals missing, it is possible to take the counterparts of the missing residuals from the neighboring frames and weight them so that the sum of the residuals become zero. In most cases, this method performs well even at relatively high loss rates for the scalefactors.

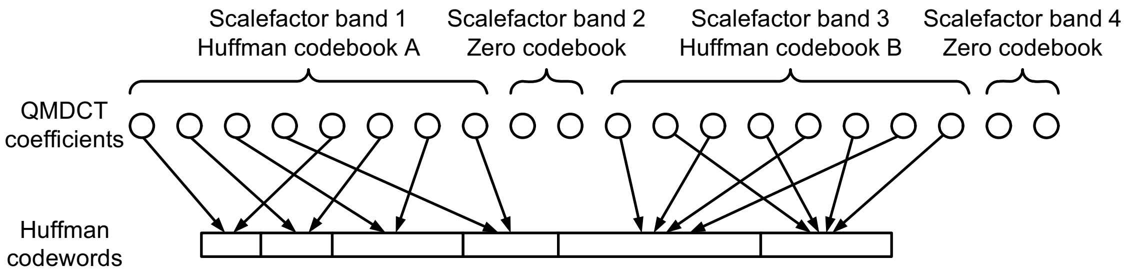
3.3.2 Coding of Spectral Samples

The conventional AAC codec applies Huffman coding also to the quantized MDCT coefficients. There is a set of predefined Huffman codebooks in the standard. The decoder selects the appropriate Huffman codebook separately for each scalefactor band according to the Huffman codebook index number given in the critical data section. This is why the Huffman coded data cannot be decoded without the critical data. Each Huffman codeword comprises two or four adjacent QMDCT coefficients, depending on the codebook. The conventional method for generating the Huffman codewords out of the QMDCT coefficients is shown in Figure 10 a). In this example, there are four scalefactor bands. The scalefactor band 1 uses the Huffman codebook A for packing two MDCT coefficients in each codeword. In contrast, the Huffman codebook B used in the scalefactor band 3 packs four spectral coefficients in every codeword. It is noteworthy that there is a special codebook index, the zero codebook. The MDCT coefficients coded with the zero codebook are all zeros, and they are actually not coded at all.

According to the rationale of interleaving, it is more harmful to lose several adjacent samples than the same number of samples separate from each other. Based on this assumption, we have implemented a simple interleaving scheme that spreads the adjacent samples within each scalefactor band into separate Huffman codewords. The scheme is shown in Figure 10 b). The concept is somewhat similar to the interleaving introduced in Transform-domain Weighted Interleave Vector Quantization (TwinVQ) audio coding method [57], except that in our system interleaving is performed separately for each scalefactor band. The performance improvement achieved with this mechanism has not been proven very significant, but the cost is low as well: notable increment in the average Huffman codeword length has not been observed when testing the scheme.



a) Conventional Huffman coding for QMDCT coefficients



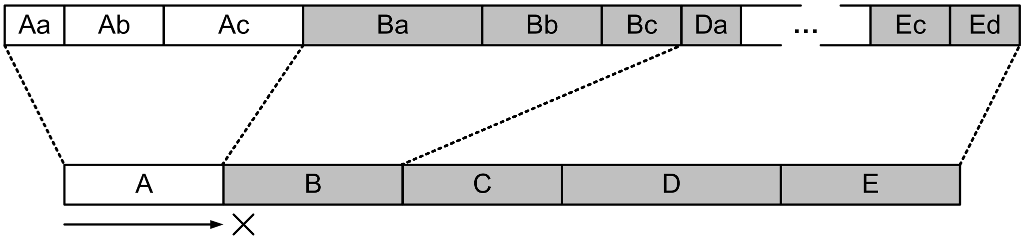
b) Proposed interleaving for QMDCT coefficients

Figure 10. Huffman coding of QMDCT coefficients illustrated: a) the conventional scheme, b) the proposed interleaving scheme.

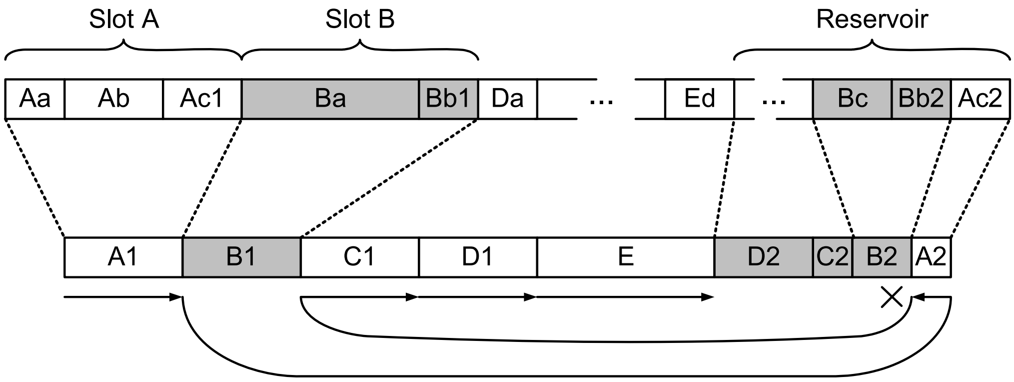
3.3.3 Huffman Code Allocation

The major problem with variable length coding is the bit error propagation. One single bit error can mutate the length of the codeword. In this case, the start position of the following codeword is wrong, leading to misinterpretation of all the following codewords. Additionally, it is impossible to skip codewords in the decoding process, because the decoder cannot know the exact length of the codewords without decoding them. This is a serious limitation for interleaving the Huffman codewords in the similar fashion as shown in Figure 7. The critical data may get lost or damaged, and in this case it is not possible to read the respective Huffman codewords. In practice, this means that the de-interleaving process cannot be continued.

One of the error resilience tools in the MPEG-4 AAC standard is the Huffman Code Reordering (HCR) tool. It aims to increase the resilience against bit errors by allocating the high priority Huffman codewords in predefined positions. The gaps left between the priority codewords are filled with lower priority codewords [23]. Inspired by this method, a robust scheme for packetizing the Huffman codewords is proposed. In our method, there is a fixed size slot reserved for each frame in every packet. The remaining Huffman codewords that do not fit in the slot are allocated in a reservoir area in the end of the packet. This approach isolates the impact of error propagation in the reservoir area.



a) Conventional allocation of Huffman codewords in packet payloads



b) Proposed robust allocation of Huffman codewords

Figure 11. Huffman code allocation illustrated: a) conventional, b) robust. Gray color shows the codewords and data sections that cannot be read due to missing critical data for frame B.

Figure 11 illustrates the proposed method. In this example, the packet contains codewords from five different frames (frames A..E). There are several codewords for each frame, marked by lowercase alphabets. Assuming that the critical data for the frame B is lost, the arrows in Figure 11 show how the decoding proceeds. All the codewords after the section B are lost, because the decoder cannot find the beginning of the section C after the section B. This is why the de-interleaving procedure cannot continue.

Figure 11 b) depicts the proposed data allocation scheme. There is a fixed size slot reserved for each frame. The overflowing codewords are allocated in the reservoir area in the end of the packet, in a reverse order. This is why the sections reserved for the frames are actually divided in two parts (A1 and A2, B1 and B2, …). Because the slot size is known by the decoder, it can now skip over B1. Only C2 and D2 are lost, because the length of B2 is not known a priori.

It is noteworthy that the slots for different frames do not need to be in the same order in every packet. In fact, it is beneficial to use reverse order in every second packet to minimize the loss of data in the reservoir in the worst case when the critical data for a frame or some frames is lost. For example, in Figure 11 b) the data sections B1, B2, C2 and D2 become useless, because the critical data for the frame B is missing. If the slots were arranged in reverse order from E to A, only the sections B1 and B2 would get lost. When the reverse ordering is applied to the half of the packets, the total number of the lost data sections would be always the same, not depending on which frame is missing its critical data. If more than one frame have lost their critical sections, even better performance might be achieved by allocating the slots in different (pseudorandom) order in each packet.

The error concealment strategies for recovering the missing spectral components have been discussed in [P3, P6]. Reasonable quality can be achieved by replacing the missing MDCT coefficients by zeros. A slight improvement is gained by interpolating the coefficients from the corresponding values in the preceding and succeeding frame. More advanced error concealment methods are not rationalized, considering the small quality improvement and the high complexity. More precise description of the error concealment procedure depends on the packetization and interleaving scheme. Because there are several alternative approaches for packetization and transport, the formal procedure for the error concealment is not discussed here.

3.4 Packetization and Transport Mechanisms

The implementation details for the packetization and transport scheme depend on several factors, especially the selected error protection method for the critical data. In addition to FEC and retransmissions alone, these two methods can be combined into a hybrid scheme employing both of them [29]. Several factors need to be considered to choose the most appropriate strategy for a certain application.

3.4.1 Redundancy-Based Error Correction

FEC-based error correction suits well for broadcast or multicast applications with no available feedback channel, or the network environments where the retransmission-based strategies are not feasible due to long end-to-end transport delays. The most straightforward method to implement FEC is to add redundant copies of the data in separate packets. In the proposed AAC streaming system, only the critical data requires strong protection, as it is vital for the de-interleaving process. Because the critical data comprises only a small portion of each frame, simple duplication of critical elements can be used without causing excessive overall redundancy overhead. In the method explained in [P1, P6] each critical data element is written multiple times in separate packets. The basic idea of this scheme is illustrated in Figure 12 (compare to Figure 7): each packet contains elements of different priority. The high priority elements are denoted by gray color in the figure.

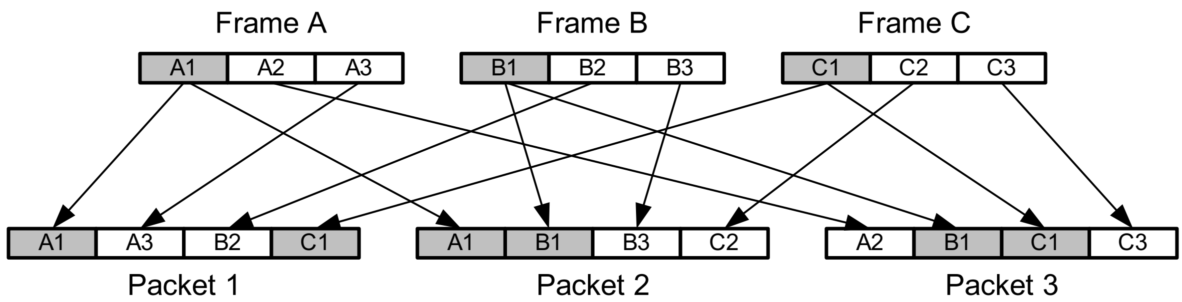


Figure 12. Interframe data shuffling with critical data replication illustrated.

The best performance is typically achieved with a long interleaving cycle, as the data losses will be spread most effectively. On the other hand, a long interleaving cycle causes a long interleaving delay. Therefore, the application has to trade-off between the interleaving delay and performance. The system must be able to recover the remaining data in each packet even if some elements are unreadable due to the loss of all the redundant critical sections for a certain frame. As long as each element has a predefined length, this is not a problem, because the decoder could skip the unreadable sections. However, this is not possible if the length of each section is variable and not known without the critical data of that particular frame. This is the case with AAC. The Huffman code allocation technique based on the fixed-size slots as presented in Subsection 3.3.3 can be used to mitigate the problem. The fixed-size slots are needed only for the frames with no related critical data present in the same packet [P3, P6].

3.4.2 Retransmission-Based Error Correction

If the required end-to-end transport delay is reasonable, the error recovery scheme based on selective retransmissions may be advantageous over FEC. In this case, RTP packets could be generated so that each data element of certain priority is packed together with elements of the same priority. This is how RTP packets can be assigned with different priority classes. If the network resources are limited, the system selects packets for retransmission according to the priority class. The basic idea of the priority-based packetization is shown in Figure 13.

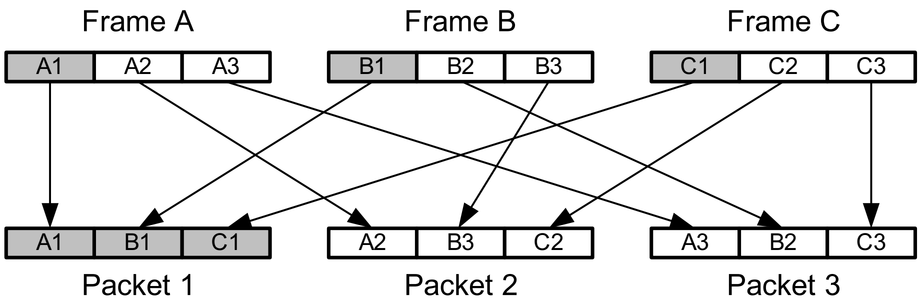


Figure13. Priority-based packetization scheme illustrated.

An important aspect is the transmission order of the packets. The selective retransmission protocol proposed in [P2] uses a decreasing order for transmission in terms of packet priority. In other words, the high priority packets are transmitted before the associated low priority packets. The main advantage of this arrangement is that there is more time allocated for the higher priority packet retransmissions. Simplified timing diagram of the protocol is outlined in Figure 14.

Another benefit is the possibility of discard intentionally one or more of the low priority packets in the end of the sequence. This is how the system can effectively control the bandwidth usage. For example, the sender may decide to drop one low priority packet per each retransmitted high priority packet. In this case, the overall transmission rate is always constant despite of the possible retransmissions. The details of the proposed approach are explained in [P2].

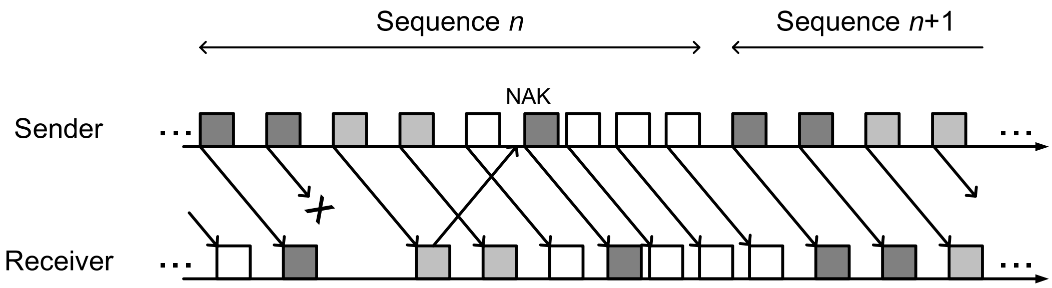


Figure 14. Example timing diagram for the proposed transport scheme. Darker color denotes higher priority.

Because the number of retransmission attempts is limited in a practical streaming application, the possibility of losing a critical packet in spite of retransmissions cannot be ruled out. In case of a critical section is lost, the associated lower priority elements are unreadable and the decoder cannot synchronize to the remaining elements beyond the unreadable ones. To avoid losing the synchronization in the low priority packets, the Huffman code allocation technique as described in Subsection 3.3.3 may be used in the lower priority packets.

3.4.3 Hybrid Schemes

There are several alternatives to employ both FEC and retransmissions in parallel in the proposed system. The most straightforward method is to use the priority-based packetization scheme as described above, but transmit each critical packet twice. A slightly more complicated method is to replicate each critical section in two different high priority packets [P6]. In this scenario, it is essential to avoid allocating more than one pair of redundant elements in the same pair of packets. Following this principle minimizes the risk of losing several critical elements in case of a pair of critical packets is lost after all. Retransmissions can be used as a backup method for packet loss recovery in the same way as explained in Subsection 3.4.2.

3.5 Application Scenarios

The proposed approach for audio streaming with prioritized packetization is especially useful when there is a system level support for the delivery of packets with different priorities. This is the case when the underlying network supports QoS differentiation or packets can be routed over links with different reliability, depending on the priority.

3.5.1 Prioritized Transmission

There are several proposals and even widely deployed technologies to support the QoS differentiation at different protocol layers, including the physical, network and application layers. In a wireless system, QoS at the physical layer allows user to transmit high priority data with more reliable means, such as using higher power in radio transmission or more reliable modulation schemes. The network layer QoS is typically based on prioritized packet scheduling in routers and gateways. In this kind of systems, the real-time packets are forwarded before the lower priority packets. The application layer schemes may allocate stronger FEC or more retransmission attempts to the priority data.

In the proposed packetization strategy, the audio stream consists of packets with different relative priorities. If the underlying communications system allows per-packet prioritization, network resources could be used more effectively, because the higher packet loss rate is allowed for the lower priority packets. Therefore, less network resources may be allocated to the lower priority packets in terms of the retransmission capacity, FEC overhead or radio power.

3.5.2 Traffic Engineering

In certain cases, there are different alternative routes available between the sender and the receiver, each with different capacity and reliability. This is the case in the peer-to-peer streaming systems, where the peer terminals are involved in routing. In this kind of environment, an intelligent system may try to optimize the perceived end user quality and network cost by transmitting the high priority data over the most reliable link with the lowest packet loss rate even if this link is more expensive. The less reliable but cheaper links could be utilized for transporting the packets of lower priority. This concept is illustrated in Figure 15.

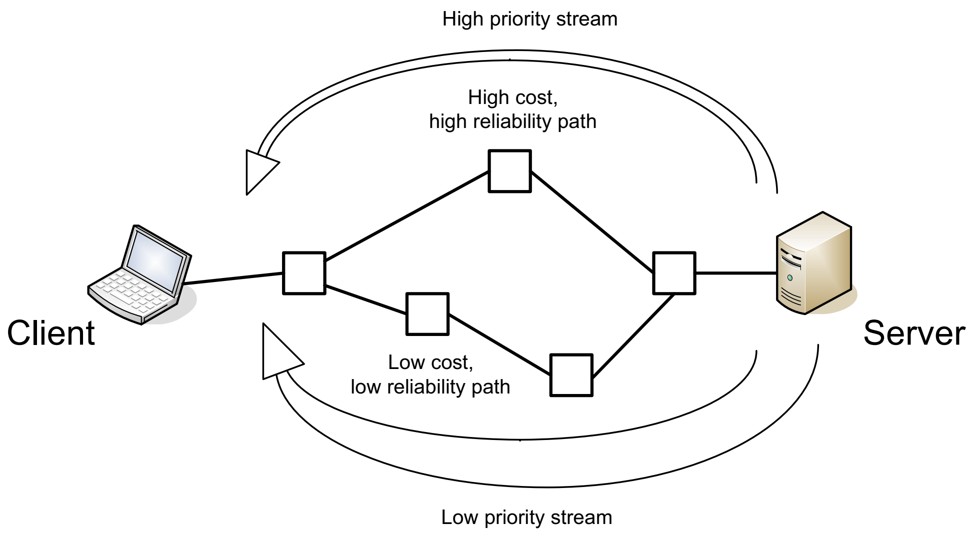


Figure 15. Prioritized streaming system using two parallel routes with different cost, capacity and reliability.

If the mobile terminal is capable of using several radio interfaces in parallel, data streams of different priority could be directed via separate radio access links. A possible use case in the future could involve a mobile terminal with two concurrent WLAN connections. The low priority data would use a free WLAN that gets congested often. In contrast, the high priority data stream would be transported through a chargeable, less congested WLAN access link. An intelligent streaming system could even dynamically change the proportion of the data directed via different routes, following the congestion conditions.

3.6 Summary

In this Chapter, application layer mechanisms for robust audio streaming have been presented. In the proposed system, each audio frame encoded with MPEG AAC is split in components of different priorities. Then, these components can be interleaved or shuffled among the transport packets. This arrangement facilitates recovery from packet losses, because each frame can be partially reconstructed even if some packets are lost. Unequal error protection mechanisms based on FEC, selective retransmissions or both could be applied to provide stronger protection to the high priority data components.

When a prioritized packetization scheme is used, each packet contains frame components of the assigned priority only. This arrangement is useful, because it allows different level of error protection to be applied for each packet, according to the priority. For example, the system could use retransmissions for the high priority packets only, or higher number of retransmission attempts could be assigned for the higher priority packets. In this case, it is reasonable to arrange packets in each interleaving cycle in decreasing order in terms of priority. This would allocate more time for the retransmissions of the critical packets. If the network occupancy is high, the system could also drop intentionally some of the low priority packets to compensate the retransmission overhead.

Although the packetization and packet loss recovery mechanisms presented in this Chapter are independent on the underlying network protocols and applicable to end-to-end communications, the proposed approach would benefit from QoS mechanisms below the application layer. For example, if the system allowed application to switch between different physical or link layer error protection mechanisms, different level of protection could be assigned to packets of different priority without using application and transport layer error management schemes, such as selective RTP retransmissions.

Chapter 4 Adaptive and Efficient Wireless Streaming

In many sense, a wireless network environment is technically more challenging than the traditional fixed IP network environments. Most of the challenges with routing in a dynamic network topology, interoperability between wired and wireless networks as well as radio resource allocation are solvable and it is possible to provide the end user with virtually the same level of easiness and reliability as with wireline IP connections. However, there are still differences between the wired and wireless connections, even observable by the end user. Not only the transmission rates are lower, but a radio link is more prone to bit errors caused by various external factors, such as fading and interference. In this Chapter, the issues related to the real-time IP transmission over a wireless link are discussed. Application layer strategies to optimize the wireless link resource utilization and the power efficiency while maintaining a high audio quality are proposed.

4.1 Bit Error Management

There are several physical factors causing distortion to radio signals, caused by the interference from other radio transmitters, multipath propagation, Rayleigh fading or the physical barriers, for example. In a digital transmission system, this kind of distortion may be observed as bit errors above the physical layer. There are several different approaches and mechanisms to recover from bit errors employed in the mainstream wireless telecommunications standards.

4.1.1 Physical and Link Layer Error Recovery

The vulnerability of a radio signal depends much on the modulation scheme and the transmission power. In addition, different link layer error management schemes can be employed to improve the error resilience. Most of the modern standardized digital wireless transmission systems include several alternative transmission modes to cope with different link conditions and quality requirements. For example, Bluetooth standard defines three error correction schemes (1/3 rate FEC, 2/3 rate FEC and ARQ) [58] and IEEE 802.11b WLAN standard provides four different bit rates from 1 Mbps to 11 Mbps [59]. There is no benefit without a cost; the use of more robust coding and modulation implies lower bit rates and poorer utilization of the radio resources.

Because the bit error rate in a digital radio channel is high in comparison to the bit error rate in cable communications, in most cases it is reasonable to use a link layer retransmission mechanism within the wireless subsystem. The link layer retransmissions are more economical than the end-to-end retransmissions, because the burden of retransmissions is distributed over the failed link only. As a radio link is typically substantially less reliable than a wireline link in spite of the link layer FEC schemes, most of the standards for wireless telecommunications use also complementary link layer retransmissions.

4.1.2 UDP Lite and Error Robustness in Multimedia Coding

Due to the link layer error detection and recovery mechanisms, there are rarely any bit errors observed at the IP or transport layer even in a wireless network environment. This is why the codec level robustness against bit errors is usually not an issue in streaming over IP. However, multimedia codecs may be used also in different applications, such as digital television broadcast or circuit-switched multimedia conferencing tools. In this kind of environments, the error resilience tools in MPEG-4 AAC and AVC may be highly beneficial.

If the multimedia codec is robust enough, discarding an erroneous packet may result in more harmful effect than passing it through the protocol stack up to the application layer. This is the rationale behind UDP Lite [19]. It is a modified lightweight version of UDP, where the area protected by a checksum is indicated in the UDP Lite header. This arrangement allows partial checksumming: only the most vulnerable parts of the packet payload are protected. Because bit errors in the unprotected area are left undetected, the packet loss rate may be significantly reduced. Figure 16 illustrates a typical checksum usage for UDP and UDP Lite.

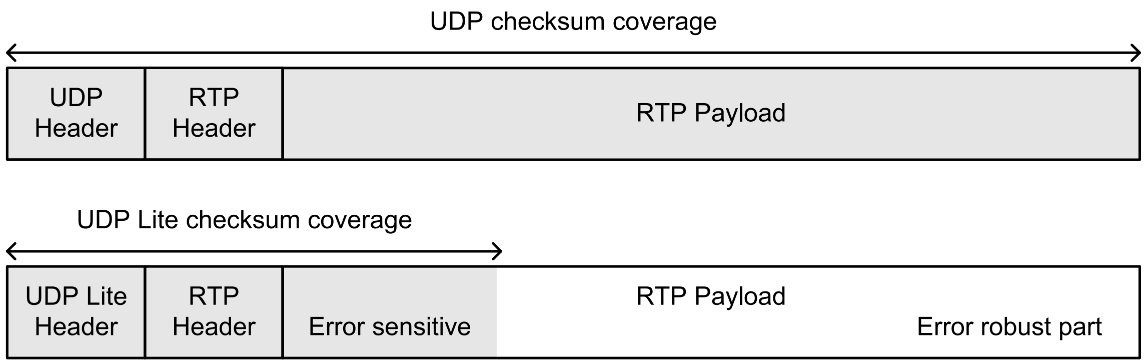


Figure 16. Traditional UDP and UDP Lite compared.

However, the advantage of UDP Lite is lost if the lower protocol layers perform functions for bit error detection and correction. As stated above, this is usually the case in wireless IP networking. This is why the practical significance of UDP Lite is marginal at the time of writing. In principle, it would be possible to deploy a link layer protocol that switches off the low layer error detection and retransmission mechanisms if the system recognizes UDP Lite datagrams. However, this would violate the conventional roles of different protocol layers.

For a more generic solution, a mechanism for cross-layer signaling would be required. For this purpose, several mechanisms have been proposed in the literature [60, 61]. Nevertheless, the activities promoting low layer protocol support to UDP Lite have been rather limited. The threshold for deploying cross-layer signaling mechanisms is high, because changes should be made at several protocol layers and the proposed proprietary signaling mechanisms do not guarantee backward compatibility in the future. In addition, the link layer FEC and retransmissions are generally considered appropriate and sufficient means for bit error management. The link layer protocols designers have traditionally had little motivation to disable the retransmissions [47].

However, there are circumstances when the use of UDP Lite would definitely be beneficial. As a first example, in a low bit rate cellular systems link layer retransmissions may cause undesirably long delays, disturbing the user interaction. Another example is multicast and broadcast, where retransmissions are not feasible. A third example is a congested WLAN hotspot. When several users are sharing the wireless resources, there is a clear motivation to use UDP Lite instead of link layer retransmissions whenever possible in order to mitigate the congestion in the shared wireless link.

4.1.3 Robust Packetization Scheme for AAC

In [P4] a robust packetization scheme for AAC streaming under conditions prone to bit errors have been proposed. The scheme is based on the idea of interleaving the elements from a group of frames into several packets as explained in the publications [P1, P3, P6], with a few modifications making it more suitable for the use with UDP Lite or any other technology that requires bit error management on the application layer.

The HCR tool of MPEG-4 AAC restricts error propagation within the Huffman coded data efficiently by allocating certain priority Huffman codewords at the known positions. However, bit errors in a wireless link are typically clustered as error bursts. This reduces the advantage of the HCR tool, because bursty errors typically damage several codewords in one frame even though the error propagation problem is partially solved. Interframe shuffling of Huffman codewords performs significantly better, because the effect of an error burst is distributed among several frames. Even very long error bursts damage only few codewords in one frame. Error propagation from frame to frame can be restricted by using fixed size slots for Huffman codewords of each frame, in the similar way as explained in Subsection 3.3.3 and [P3, P6].

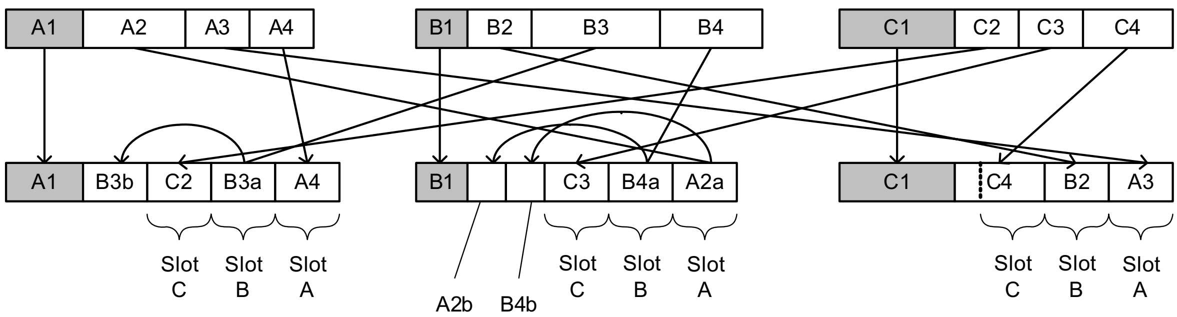


Figure 17. Bit error robust packetization scheme illustrated. Critical elements are shown in gray.

However, the packetization scheme described in Section 3.3 cannot be used with UDP Lite as such. Most importantly, the critical data must be allocated in the beginning of each RTP payload in order to be able to protect the critical part with the partial checksum. Application layer checksums could be used to detect errors in the critical parts. However, this option would lead to undesired overhead due to the bits reserved for the checksums. This is why we have allocated the critical sections always in the beginning of each payload. The fixed-size slots containing the Huffman codewords have been allocated in a reverse order, starting from the end of the payload. This arrangement allows the decoder to start parsing the Huffman codewords for different frames before reading the critical data associated to another frame first. The reservoir area is situated between the critical data and the fixed-size slots. The modified scalefactor coding scheme as described in [P3] have been employed to avoid error propagation in the DPCM coded scalefactor data. The basic idea of the packetization scheme is depicted in Figure 17.

Depending on the average RTP payload size, and the average length and distribution of the error bursts, packets may still be discarded due to errors in the protocol headers or the critical section of the RTP payload covered by the partial checksum. In this case, traditional means, such as retransmissions and FEC for the critical data, can be used for error recovery. This topic has been discussed briefly in [P4].

4.1.4 Evaluation of the Proposed Scheme

The proposed packetization scheme has been evaluated by running a test implementation in a simulated wireless environment. The simulator uses error pattern files including bit errors collected from a real WCDMA channel and applies them to the UDP datagrams to generate errors. Because there was no real UDP Lite implementation available, the functionality of UDP Lite was simulated by recalculating the conventional UDP checksum in the wireless network simulator immediately after the bit errors were applied to the datagram. If the area protected by the partial checksum is damaged, the simulator discards the whole datagram. The test environment is outlined in Figure 18. The details of the test procedure and the results are explained in [P4].

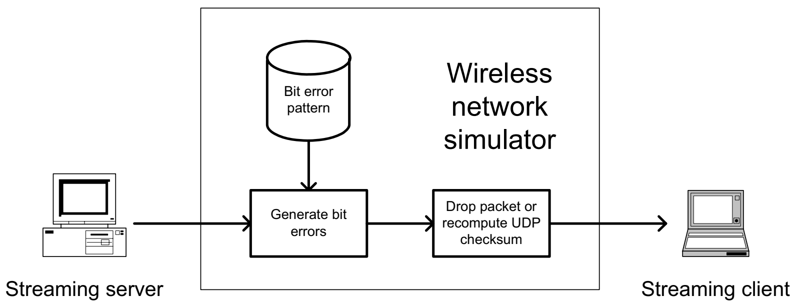


Figure 18. The wireless network simulator used for evaluating the proposed packetization scheme.

The subjective audio quality was evaluated for four codecs (MPEG-1/2 Layer II, MPEG-1/2 Layer III, baseline AAC, proposed modified AAC) and four different proportional UDP Lite checksum protection levels from 0% up to 100%. Larger checksum coverage implies higher packet loss rate. On the other hand, lower coverage implies higher probability of bit errors to occur in the RTP payload. The test was repeated with two different error pattern files.

According to the results, MPEG-1/2 Layer II is the most robust against bit errors among the baseline codecs. However, it provides the lowest audio quality in error-free circumstances. MP3 and AAC perform very poorly when there are bit errors present. The proposed modified AAC streaming systems outperforms the baseline codecs clearly when the checksum coverage is 10% or 20%. At the 50% checksum coverage, the difference is not that clear. As expected, at the 100% checksum coverage, only packet losses occur, and all codecs perform approximately equally.

4.2 Analysis of the Bit Error Characteristics

Wireless systems use different link layer technologies to cope with bit errors. The most straightforward method is to discard all the damaged packets and let the transport layer protocols to perform the error recovery functions, such as retransmissions. More advanced systems use link layer retransmissions. This reduces the retransmission delay and the overall burden substantially compared to the transport or application layer retransmissions. This is because the retransmissions take place only in the link where the packet was lost, not in all the links between the sender and the receiver. Even if the link layer error recovery mechanisms conceal the bit errors from the application layer, it is possible to gain some information about the underlying conditions by analyzing the observed packet loss and delay characteristics on the application layer.

4.2.1 Bit Errors in Wireless Links

The traditional use of UDP Lite as described in Section 4.1 is not the only possibility to benefit from the partial error detection in a wireless channel. Another potential advantage of allowing bit errors to traverse up to the application lies in the possibility to analyze bit errors at the application layer. An intelligent application takes a different approach to combat packet losses, depending on whether the losses are caused by congestion or corruption. This is why packet loss differentiation has been studied extensively [46, 62, 63].

One solution is to turn off the bit error detection mechanisms at the lower protocol layers. If the bit errors are detected at the transport control layer, the system knows that the prevailing network problems are likely to be derived from bit errors instead of congestion. This information is useful for the intelligent congestion control algorithms [46]. At the time of writing, serious effort is put in the development of Datagram Congestion Control Protocol (DCCP), supporting an optional checksum for the payload [47, 64]. This feature is primarily intended for differentiation between congestion and corruption, but it allows erroneous frames to be passed through to the application layer in a similar manner as if UDP Lite is used.

A system level approach for checksum-based adaptive streaming has been outlined in [P5]. Unprotected UDP datagrams, either using UDP Lite or traditional UDP with checksums turned off, may be used to diagnose the network conditions. In this case, application layer checksums can be used to detect bit errors. We have proposed a simple mathematical model to describe the bit error probability p as a function of the section length of l bits, as shown in Equation (1) [P5, P7]:

$$p(l) = 1 - c \cdot \exp(-\lambda l) \tag{1}$$

In Equation (1), the average proportion of erroneous bits is 1-c and the average number of bits between two bit error bursts is $1/\lambda$. Therefore, assuming that c remains the same, a large value of λ suggests that there are a lot of short error bursts, whereas a small value of λ means that the error bursts occur more rarely but their average length is longer. The parameters c and λ can be solved from Equations (2) and (3), respectively, if the observed error rates are known for the sections of two different lengths (l_1 and l_2) [P7]. Therefore, it is possible to gain more precise information on the distribution of bit errors by dividing each packet in two or more separate sections covered by different checksums [P5].

$$\lambda = \ln\left(\frac{1-p(l_1)}{1-p(l_2)}\right) \Big/ (l_2 - l_1) \tag{2}$$

$$c = \frac{1-p(l_1)}{\exp(-\lambda l_1)} = \frac{1-p(l_2)}{\exp(-\lambda l_2)} \tag{3}$$

4.2.2 Packet Losses in Wireless Links

In an error prone radio channel, the packet failure rate depends on the packet size, because large packets are statistically more likely to be hit by bit errors than small packets. On the other hand, a small packet size implies higher header overhead. This is why packet size optimization has been studied extensively. Research in this field has been mainly focused to achieve the optimal throughput in different system architectures and error conditions. During the past decade, several different approaches have been proposed for packet size adaptation under dynamic network conditions [65-67].

Most of the packet size optimization schemes use link layer packet fragmentation or aggregation to achieve the optimal packet size. However, this kind of strategies do not benefit from the advantages of smart application level framing, such as allocating an integer number of media frames in each packet. If the media frames were fragmented arbitrarily, loss of one fragment could make all the related fragments useless. In addition, fragmentation causes extra delays that could be avoided by using smaller packet size at the application layer. On the other hand, packet aggregation at the link layer may increase the probability of bursty packet losses. Application layer packet size optimization could be used for priority differentiation: the high priority packets could be made smaller to decrease the loss probability in proportion to the lower priority packets.

The packet size optimization for streaming applications has been discussed in [P5, P7]. When the bit error probabilities are assumed to follow equation (1) and the erroneous packets are discarded by an error detection mechanism, we can derive the connection between the packet length and the packet loss probability. Two examples are shown in Figure 19: in the presence of a lot of short bit error bursts the curve is steeper compared to the case of long error bursts at lower density. In the former case, packet size optimization is especially useful.

If there were no retransmissions involved and the bit error rate was known, it would be possible to adjust the packet size so that the required maximum allowed packet loss rate is exactly met. For example, the packetization strategy explained in Chapter 3 allows relatively flexible adjustment of the critical, intermediate and low priority packet payload sizes. This may be done by changing the relative amount of packets used to carry data of each priority. Typically, it is rational to make the critical packets smaller than the other packets by allocating more packets for critical data than the relative proportion of the critical data would suggest. If retransmissions are used, each packet loss causes an extra burden in the network load in the form of retransmission overhead. In this case, the proposed model can be used to estimate the optimal packet size providing the lowest network resource utilization.

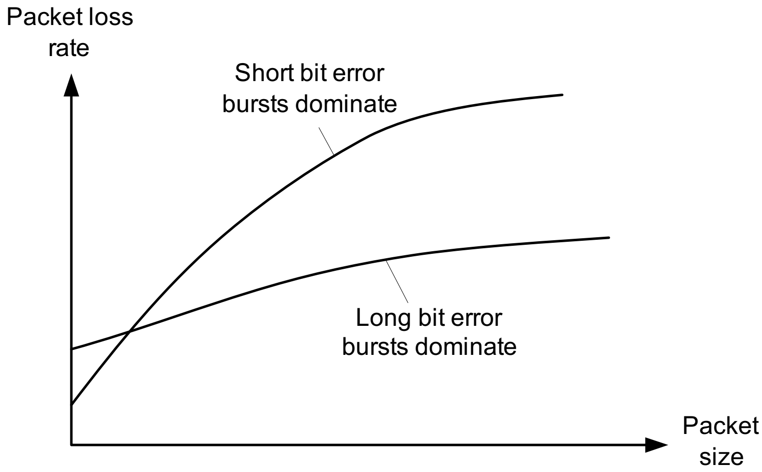


Figure 19. The relationship between the packet length and the packet loss rate under the presence of bit errors following different patterns.

4.2.3 Delay Characteristics in Wireless Links

Link layer retransmissions are commonly used complementing FEC and robust modulation schemes in many wireless telecommunications systems. In application developers' point of view, the link layer error correction and recovery mechanisms seem to make packet size optimization obsolete. However, even the best link layer error recovery strategies cannot entirely conceal the impact of transmission errors from the upper layers. The strong FEC schemes increase the redundancy overhead in the transmission channel, reducing the bandwidth efficiency and increasing the transport delay of each packet. Retransmissions reduce the observed packet loss rate on the application layer, but may have a major effect on the end-to-end delay characteristics. Each time a corrupted packet is retransmitted, an additional delay is introduced. Some packets are received correctly at the first attempt, whereas several retransmission attempts may be needed to transport some others. This is why the average delay and jitter may be increased substantially.

As the link layer retransmissions are taking place in the erroneous link only, the additional delay is usually negligible in a WLAN environment, where the transport delay in the radio link is usually very short. In the real-time communications over a low bit rate cellular radio link, the retransmission delays would play a bigger role as a potential cause of problems. However, retransmissions always degrade the overall bandwidth efficiency. As stated above, large packets in a radio channel are more likely to be hit by bit errors and thus be retransmitted than small ones. Therefore, there is a clear motivation for packet size optimization even if the packet losses are recovered via link layer retransmissions. This point of view should be emphasized especially when the network bandwidth is limited or there are a lot of users sharing the wireless resources. This is often the case in the cellular networks and WLAN hotspots.

In [P7], the relationship between the packet length and the end-to-end delay distribution has been studied. Each retransmission adds an extra delay to the end-to-end packet transport time. It is assumed that the retransmission delay is constant in practical accuracy, although there is some random variance in practice. Therefore, we can expect the cumulative distribution of the relative end-to-end packet transport times to follow roughly the shape shown in Figure 20. In this example, the packet loss rate is 50%. The mathematical formulation of the hypothesis would be as follows: the probability P that the observed packet transport delay time t_{OBS} is smaller than t follows Equation (4), where p is the packet loss rate in the wireless channel, t_0 is the initial transport delay and t_r is the retransmission delay.

$$P(t_{OBS} \leq t) = 1 - p^{(t - t_0 + t_r)/t_r} \tag{4}$$

Because every packet contains a timestamp, a streaming application can easily keep track on relative one-way trip times and approximate the actual shape of the packet arrival time distribution function. Assuming that the distribution follows roughly the theoretically derived distribution, the average retransmission delay t_r and the packet loss rate p can be approximated by solving them from the Equation (4) with some suitable sample values for t and t_{OBS}.

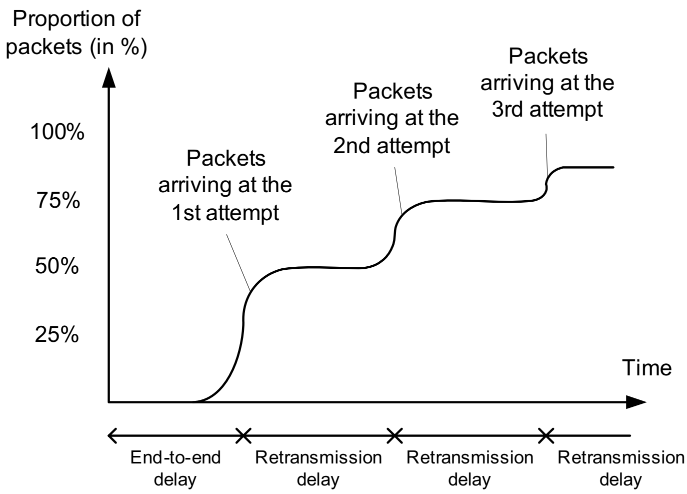


Figure 20. The theoretical cumulative distribution function of the packet arrival times, when the packet loss rate is 50%.

When this method is applied to solve the packet loss rate p for packets with different lengths, it is possible to estimate the parameters c and λ describing the features of the typical bit error patterns in the channel as explained in Subsection 4.2.2. This information can be used by the application to select appropriate packet sizes and modes of transmission, as explained in [P5]. In demanding wireless networks and difficult radio link conditions, this can be beneficial in order to optimize the network resource utilization and the quality experienced by the end user.

The practical experiments described in [P7] were performed using two laptops with IEEE 802.11b compliant built-in WLAN cards communicating in the ad hoc networking mode. The power-save features were disabled and the ad hoc mode was used in order to eliminate the impact of the additional delays caused by the power-save mechanism and queuing in the backbone network. The results support the hypothesis derived from the theoretical analysis.

In fact, minimizing the link layer retransmissions is not beneficial only because of the network resource utilization. Reduced radio link activity improves power efficiency as well. As the battery technology remains one of the most significant problems in mobile computing, power efficient wireless networking is one of the most important challenges in wireless multimedia streaming. If a bursty transmission mode is used to enable better power efficiency as explained in Section 4.3, timing would play a major role in packet delivery. In this case, packet size optimization could be especially helpful for the system allowing it to achieve the maximum power efficiency.

In practice, the total path delay includes not only the delay in the wireless link, but also the queuing and transport delay in the fixed part of the network. In some cases, such as in a high bandwidth LAN environment with little contending traffic, we may assume that the wireline delay is nearly constant and it is included in t_0. Queuing causes some random variation in the observed total path delay, but it is typically not related to the packet length. Therefore, it is often possible to analyze the differences in transport delays for the packets of different length, even if there is little knowledge of the network architecture behind the wireless link. In general, the more routers and links there are between the sender and the receiver, the less reliable results concerning the conditions in the wireless link can be derived by analysing the transport delays.

4.3 Power-efficient Streaming based on Bursty Transmission

The bursty transmission mode allows long periods in the sleep mode between two bursts, decreasing the power consumption characteristics substantially. In theory, long bursts provide better power efficiency, because the gaps between bursts are larger. On the other hand, long bursts are more likely to cause congestion, because the periods of high transmission rate are longer. Therefore, selecting the appropriate burst length is not a trivial task. This issue is addressed in [P8].

4.3.1 Impact of Burst Length and Transmission Interval

The relative power saving is the proportion of time spent in the sleep mode during each burst cycle (t_{sleep}/t_{cycle}). If we assume that there is no variation in the end-to-end transport delay and the receiver can accurately predict the arrival time of the first packet in a burst, it can be computed from the equation (5). The parameters needed are the required average media data rate B, the average payload length P, the number of packets in each burst n, the transmission interval (gap) between two packets t_{gap} and the transition period when switching between the active and sleep modes t_{on_off}.

$$\frac{t_{sleep}}{t_{cycle}} = 1 - \frac{B(t_{gap}(n-1) + t_{on\_off})}{nP} \quad (5)$$

In practice, the situation is usually more complicated. A short gap between two packets imply that the peak transmission rate during the bursts is high. If the momentary transmission rate is close to the maximum capacity of the link, the packet loss and high jitter probabilities increase, especially in the end of each burst. If there is contending traffic present, bursty transmission can harm the overall network performance and disturb the other users significantly as well. Apparently, the optimal burst length is a trade-off between power efficiency and fair use of the network capacity.

Practical experiments have been conducted in a real WLAN environment to find out the suitable gapping and burst length in different network conditions. A regular desktop PC was used as a test server and a laptop connected to a WLAN infrastructure as a test client. The packet losses and the relative transport delays were stored in a log file at the test client and analysed after the experiments. Another heavy data flow was generated between the streaming server and an additional "dummy" mobile client to cause contending traffic in the WLAN access point and the shared radio medium. The physical test arrangements are illustrated in Figure 21. The tests were performed in the campus network of the National University of Singapore, using Cisco Aironet WLAN Access Points and laptops with Intel's IEEE 802.11b compatible built-in WLAN cards. The streaming experiments were repeated with different gappings, burst lengths and congestion conditions [P8].

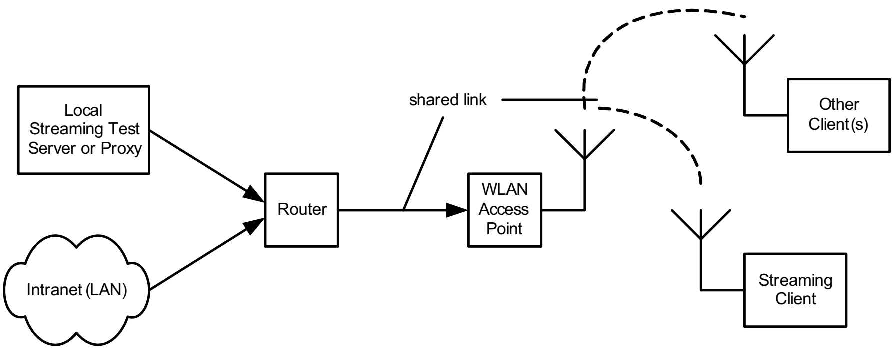


Figure 21. Network diagram of the test arrangements.

According to the results in [P8], transmission interval does not seem to have a major influence on the observed packet loss and jitter characteristics. In contrast, the impact of the burst length is significant if there is contending traffic present. If the network and link occupancy is low, long bursts can be used without substantial problems in the form of increased packet loss rate or jitter.

4.3.2 Adaptive Burst Length

As the experimental results show, the optimal burst length varies according to the network conditions. To overcome this controversy, the streaming system should select the appropriate burst length dynamically for different network conditions. A test system was implemented to monitor the packet loss rate and jitter, and change the burst length according to the changes in the observed network characteristics [P8]. Performance of the test system was evaluated by experiments in the same environment that is described in Subsection 4.3.1.

The algorithm for burst length adaptation is rather simple and relies on three basic principles:

1) High packet loss rate indicates severe congestion. If the packet loss rate increases suddenly, system switches immediately to the non-bursty (conventional) transport mode.

2) If the short-term jitter gets substantially higher than the average long-term jitter, the burst length is reduced.

3) If the short-term jitter gets substantially lower than the long-term jitter, the burst length is increased unless the maximum burst length is reached already.

The basic principle of the jitter computations is similar to that explained in the RTP specifications [49]: jitter j is computed iteratively with the smoothing function (6), where D_i is the difference between the observed arrival time and the transmission time (timestamp) for packet i, and the smoothing factor α (0..1) defines the sensitivity of the function. For the long-term jitter, larger value of α is used to make the distinction.

$$j_i = \alpha j_{i-1} + (1-\alpha)(|D_i - D_{i-1}| - j_{i-1}) \tag{6}$$

However, because the jitter is typically much larger for the last packets within a burst, the jitter computations in the test system are not based on the difference of the relative arrival times between two adjacent packets, but between the first and the last packet in each burst. Therefore, the Equation (6) becomes Equation (7), where n is the burst length, D_1 is the difference between the timestamp and the receiving time for the first packet and D_n for the last packet in the burst, respectively.

$$j_i = \alpha j_{i-1} + (1-\alpha)\frac{(|D_n - D_1| - j_{i-1})}{n} \tag{7}$$

According to the experimental results, a streaming system using adaptive burst length is highly beneficial if there is occasional congestion present. A slightly higher power saving rate can be achieved than if using always the maximum burst length. In terms of the packet loss rate the benefit is even more significant, especially if the lower priority packets in the end of each burst suffer substantially lower average loss rates when the adaptive burst length is used instead of a fixed burst length [P8]. However, there is still a lot of work to optimize the adaptive streaming scheme for different wireless environments and network conditions.

4.4 Application Scenarios

Most of the research work included in publications [P4, P5, P7] does not have direct applications in today's commercially available wireless IP systems. There are very few, if any, commercially deployed systems that allow delivery of erroneous packets up to the transport layer so that the use of UDP Lite would be justified. However, the raise of the research and development work in cross-layer optimization may change the situation radically in the near future [47, 64]. Especially the development of DCCP may catalyst the efforts in deployment of the cross-layer optimization mechanisms. Even without cross-layer optimization, the proposed concepts can be used to optimize the network resource utilization and the end-to-end transport delay.

4.4.1 Bit-Error Resilience

One relevant aspect in cross-layer optimization is the conveyance of erroneous packets to the application layer. Several studies suggest that the use of UDP Lite or a comparable protocol may substantially improve the capacity of a multiple access radio network [68, 69]. This is why the delivery of corrupted data through the protocol stack may become an option supported by several wireless standards in the future. The proposed packetization scheme in [P4] provides good bit error robustness, making it very beneficial in wireless systems with an option of partial checksumming.

It may be especially useful to apply the proposed scheme for broadcast transmission systems, because broadcasting typically lacks a feedback channel and retransmission-based error recovery is not a feasible option. Mobile television is a promising new application, and because broadcasting in IP networks is not ubiquitous, complementary technologies are needed for mobile TV broadcasting services. This is why mobile terminals compatible with digital broadcast standards, such as DVB-H, are likely to get popular in the near future. DVB-H uses IP-based encapsulation of data and FEC for bit error correction [70], and it would clearly benefit from robust media packetization.

4.4.2 Wireless-Aware Adaptive Streaming

The conventional streaming applications are designed for fixed IP network topologies and typically they do not achieve the best possible performance in wireless systems. Especially, the conventional measures for congestion control may not be optimal in the case of errors appearing in a wireless channel. Instead of reducing the transmission rate, the wireless packet losses might be better combated using UDP Lite or reducing the packet size.

There are several alternatives to cope with different network conditions [P5]. Transmission rate adaptation is the best and often the only choice in case of congestion. If the packets are lost due to bit errors but the low layer error detection mechanisms cannot be turned off, the packet loss rate can often be controlled by using a smaller packet size. In this case, there is no need to restrict the use of selective retransmissions. Depending on the underlying technology, large packets may suffer from longer end-to-end delays because of the link layer retransmissions. In this case, it is sometimes useful to limit the packet size for the most time-critical data.

If the error detection can be omitted and UDP Lite can be used, there are more options for the error recovery and streaming adaptation at the application layer. In this case, the proper strategy depends on the bit error distribution. Long error bursts cannot typically be recovered using weak bit level FEC as easily as short error bursts or individual bit errors. This is why the long bit error bursts can usually be combated best by using small packet size and normal transport layer error detection together with packet level FEC or selective retransmissions.

In contrast, when short bit error bursts are dominating, it is more beneficial to let bit errors pass up to the application. The individual bit errors or short errors bursts can often be recovered at the codec layer using error correcting codes, robust source coding or bit-error tolerant packetization strategies as explained [P4]. Another option is to use partial retransmissions [71, 72]. There are several implementation alternatives for partial retransmissions, but the basic idea is simple: the payload is divided in several slots and the errors are detected in each slot separately. Therefore, the receiver can choose to retransmit only the damaged slots. If a large packet size is used and only a small portion of the packet is hit by bit errors, this strategy may reduce the retransmission overhead significantly.

In our contribution [P5], we outline the decision-making process to choose the optimal strategy in different circumstances. In general, the principles summarized above are followed. A decision tree for the process is shown in Figure 22. In [P5] the primary focus is in the strategies using unprotected datagrams. When the loss rates for the unprotected and the protected datagrams are compared, it is possible to find out whether most of the packet losses are due to bit errors or congestion and decide the appropriate action according to this information. If it is not possible to disable the error detection (protection), the only available functions for adaptation on the application level are transmission rate adjustment and packet size optimization.

In adaptive streaming, the role of the feedback information is essential. When RTP is used for the data transport, it is easiest to use application-specific RTCP messages to convey the relevant statistical information and even the retransmission requests. Although RTCP reports are intended to be sent periodically, the critical RTCP messages can be advanced to ensure timely feedback. If the distance between the server and the client is long, the round-trip time could be too long for efficient delivery of the critical messages, especially the retransmission requests. In this case, it would be appropriate to employ an application-aware proxy as close to the last (wireless) link as possible. In this way the feedback loop for time-critical messages could be limited between the proxy and the mobile terminal.

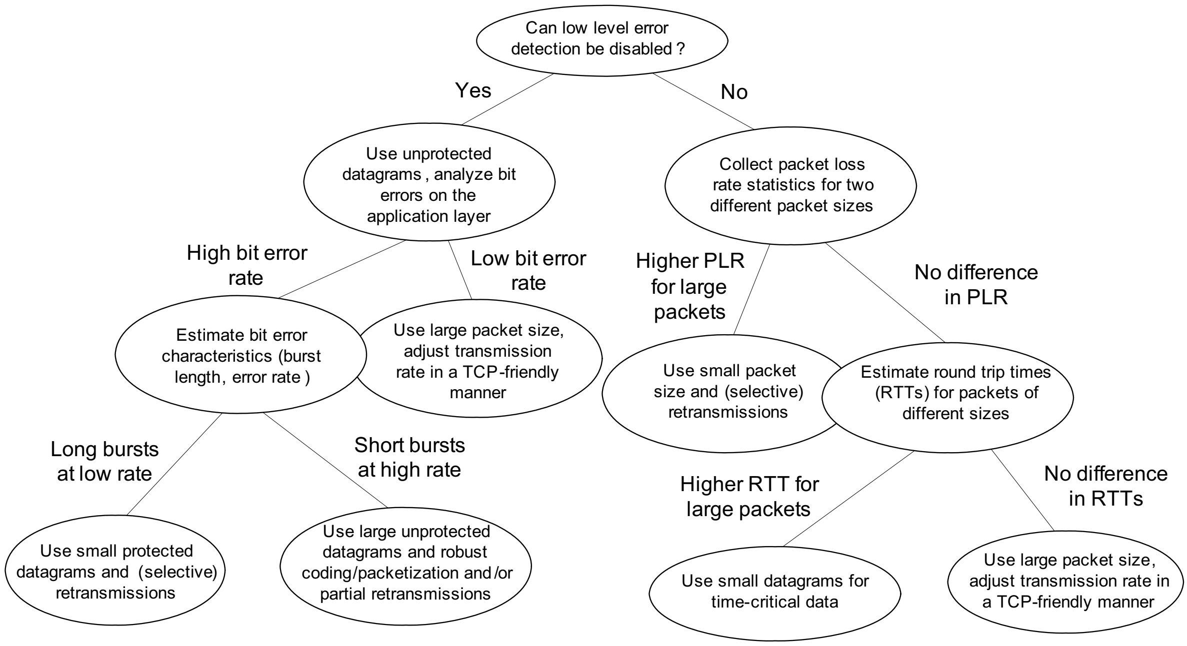


Figure 22. The decision tree for a streaming application to select between different strategies to cope with different network conditions.

4.4.3 Power-Efficient Adaptive Streaming

Power-aware streaming is another form of wireless-aware streaming. As explained in Section 4.3, the use of bursty transmission mode with adaptive burst length in multimedia streaming may substantially improve the power efficiency. Figure 23 shows how a prioritized sequence of audio packets could be turned into bursts of different length. It is essential that the packets of different priority are distributed evenly among the bursts and the packets are arranged in decreasing order in terms of priority within each burst. When the packetization and transport guidelines such as explained in [P3] are used, it is rather simple to design an adaptive multimedia streaming framework that supports the co-existence of the different wireless-aware adaptive streaming techniques explained in this Chapter. The signaling could be handled using RTCP or a proprietary signaling protocol in a similar manner as explained in Subsection 4.4.2.

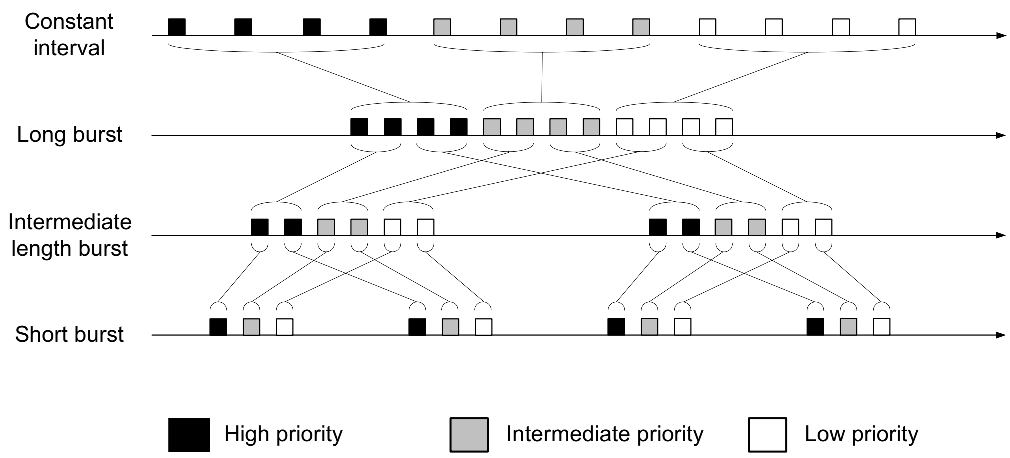


Figure 23. Generation of packet bursts with different lengths.

There are several reasons why the transport mode described in Subsection 3.4.2 would be useful for the bursty transmission. Firstly, when each interleaving cycle is transmitted as a burst or a group of bursts, the interleaving latency is minimized. Secondly, the packet loss probability for the high priority packets is lower in the first place, because they are allocated in the beginning of each burst. This is an important benefit especially if the selective retransmissions cannot be used.

4.5 Summary

In this Chapter, different technical solutions improving the performance of multimedia streaming in a wireless environment have been presented. The primary difference between a wireless and wired link is the typically higher occurrence of bit errors in a wireless channel. Therefore, bit error management plays a major role in wireless telecommunications. Many wireless technologies utilize link layer error management techniques to conceal the bit errors from the upper protocol layers. Nevertheless, some media decoders tolerate a reasonable amount of bit errors. In this case, the wireless link utilization could be significantly improved by allowing erroneous packet to be conveyed to the application instead of dropping or retransmitting them. We have proposed an interleaving and packetization scheme for MPEG AAC improving the robustness to bit errors significantly.

However, most of the practical wireless access technologies do not let the user to switch off the link layer error recovery mechanisms. Even in this case, a carefully designed packetization method may improve the wireless link utilization and the overall network performance. Apparently, large packets are more likely to be hit by bit errors, leading to high retransmission overhead. On the other hand, the header overhead is higher for the small than the large packets. We have studied the relationship between the packet size and observed retransmission delays under different bit error conditions and proposed general guidelines for packet size optimization under different bit error characteristics.

Besides error management, power efficiency is also an important issue in the mobile telecommunications. The power saving schemes of the dominant wireless standards are typically ill-suited for streaming applications receiving a steady flow of packets instead of occasional packet bursts. Better power saving ratio can be achieved by reshaping the packet flow in bursts, when the mobile terminal can spend more time in the sleep mode between the bursts. Unfortunately, bursty transmission requires a high peak transmission rate, which increases the risk of congestion to occur. We have proposed an adaptive approach, where the burst length is selected according to the prevailing network conditions. With this method, a reasonable trade-off between the power saving ratio and congestion avoidance can be achieved.

Basically, all the mechanisms proposed in this Chapter are intended to reside at the application layer. However, only the robust packetization scheme suits well for end-to-end communications when the sender and the receiver are located in different IP subnets. The proposed methods for bit error analysis, packet size optimization and power-efficient traffic reshaping perform best when the total number of links before the wireless link is minimized. This is why the use of application-aware proxies located near the wireless access link is preferable, at least if the distance between the original sender and the receiver is long. Theoretically, more reliable results could be achieved by analyzing the bit error characteristics at the link layer. In this case, the information about the error characteristics should be conveyed from the link layer up to the application that is responsible for the adaptation procedure. Unfortunately, cross-layer communications are poorly supported in today's practical protocol architectures.

Chapter 5 Summary of Publications

This Chapter summarizes the publications incorporated in this dissertation and describes the author's contribution to the publications. The publications can be roughly divided in two separate, but slightly overlapping modules. Publications P1, P2, P3 and P6 describe the proposed advanced system for packet loss resilient audio streaming. Publications P4, P5, P7 and P8 comprise a less tight module studying different approaches for efficient multimedia streaming in challenging wireless network environments. Figure 24 illustrates the workflow and the inspirations between the separate publications.

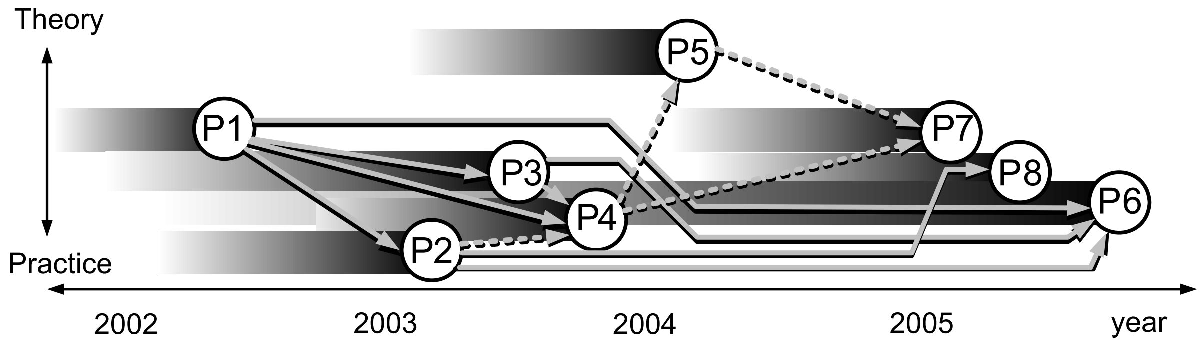


Figure 24. Rough workflow and the relationship between the publications included in this dissertation.

5.1 Overview of the Individual Publications

5.1.1 Publication 1

In this publication the idea of achieving high robustness against packet losses by spreading the frequency components of a single perceptually coded audio frame into several packets was first proposed. It was shown that the critical sections of each frame could be protected efficiently with simple FEC using data repetition. The total redundancy overhead is reasonable, because only the critical data requires strong protection, and its proportional amount is small.

5.1.2 Publication 2

In this publication a transport strategy for audio streaming based on selective retransmissions has been introduced. The proposed scheme separates different data components from each AAC frame and shuffles them among packets with respective priority classes. Using this method, a group of AAC frames with similar priority can be turned into a set of packets with different priorities. The streaming server arranges each set of packets so that the packets with highest priority are transmitted first. This allows more time for retransmissions compared to the packets of lower priority. The system can even adjust the bandwidth consumption by deliberately dropping some low priority packets in the end of each packet sequence.

5.1.3 Publication 3

This publication extends the work published in the publications 1 and 2. Most importantly, the original scalefactor coding method of AAC based on DPCM has been replaced to avoid error propagation if there are missing individual scalefactors. Another significant improvement is the Huffman code allocation scheme in the packetization procedure.

5.1.4 Publication 4

In this publication a bit error tolerant packetization strategy for audio streaming has been presented. The work has its foundation in the concepts presented in the publications 1 and 3. The basic concept has been modified to allocate the critical data always in the beginning of each payload. This approach allows critical sections to be covered by partial UDP Lite checksums. The proposed scheme has been tested by applying realistic bit error patterns to the test payload. The subjective analysis show significant improvement in audio quality compared to the baseline codecs, such as conventional AAC and MP3, when the bit errors are introduced to the payload.

5.1.5 Publication 5

In this publication a concept of adaptive streaming in a heterogeneous network environment has been proposed. Different strategies have been suggested to cope with the bit errors and packet losses, depending on the type of network problems and the bit error patterns. The proposed system uses unprotected datagrams, either UDP Lite or conventional UDP with checksumming disabled, together with application layer checksums to analyze the bit errors in the channel. If the packet losses are related to congestion rather than corruption, conventional TCP-friendly congestion control may be used. Otherwise, the system may select between more suitable strategies, such as partial retransmissions, error correcting codes and packet size optimization.

5.1.6 Publication 6

In this publication the work published in the publications 1, 2 and 3 has been put in a larger context. The paper summarizes the concept of improving robustness of streaming audio by spreading the internal elements in each AAC frame over several RTP packets. Different methods to protect the critical data sections (FEC, retransmissions and hybrid) have been studied and compared analytically, and a slightly improved version of the robust packetization scheme introduced in [P3] has been proposed. As one of the major contributions in this paper, the formal listening test results justify the proposed approach of shuffling the frequency components among several packets.

5.1.7 Publication 7

In this publication the impact of the packet size to the observed end-to-end transport delay, jitter and packet loss rate in a wireless environment has been studied both theoretically and experimentally. A simple mathematical model has been derived to model the relationship between the packet size and the relative transport time distribution when the characteristics of the prevailing bit error patterns (bit error density and burstiness) are known. This approach allows the application to gain information about the underlying channel conditions by analyzing the jitter. The hypothesis based on the theoretical study has been verified by experiments in a real WLAN environment.

5.1.8 Publication 8

In this publication, power-efficient streaming based on bursty transmission mode is studied. First, the proportional power saving and packet loss rates with different parameters (burst length and packet transmission interval) have been studied analytically and experimentally in a real WLAN environment. Theoretically, better power save rates can be achieved with longer bursts, but long bursts may also cause more severe congestion-related problems, such as increased packet loss rate and jitter. The experimental results support the hypothesis that the packet loss rate is higher for the last packets within a long burst especially if the congestion conditions are difficult. As a solution to this dilemma, a system with a dynamically adjusted burst length has been proposed. The preliminary experiments show that this approach provides a good trade-off between transport reliability and power-efficiency under fluctuating network conditions.

5.2 Author's Contribution to the Publications

In publication 1, the author proposed the concept of spreading the components of each frame among several packets. He also implemented the proposed scheme in a real-life streaming framework and conducted the experiments reported in the paper. The author is the sole writer of the paper.

In publication 2, the author proposed the transport strategy based on selective retransmissions and decreasing transmission order of packets in terms of priority to allocate more time for retransmissions of the critical packets. He also implemented the scheme and analyzed the bandwidth consumption of the proposed system via a theoretical analysis and practical experiments. The author is the sole writer of the paper.

In publication 3, the author proposed the modified scalefactor coding and improved Huffman codeword allocation to overcome the practical problems in the streaming system described in the publications 1 and 2. He also programmed the proposed schemes in the practical streaming framework.

In publication 4, the author designed and implemented the packetization scheme for the bit error robust streaming, mostly relying on the previous work described in the publications 1, 2 and 3. The streaming tests in the simulated network environment were carried out by Mr. Roope Järvinen, and the results were analyzed by joint efforts of both authors.

In publication 5, the author proposed the concept of using unprotected datagrams in order to gain more precise information on the channel conditions and using this information for streaming adaptation. He also made the related literature study on the topic. The author is the sole writer of the paper.

In publication 6, the novel techniques for the streaming system described in the publication, mostly based on the ideas presented in the publications 1, 2 and 3, have been primarily designed and implemented by the author. He also initiated the efforts to verify the proposed streaming concepts via listening tests and produced the audio samples for tests. He also performed the theoretical analysis on the residual frame loss rate and bandwidth consumption when different techniques to recover from packet losses are used. The formal listening tests were arranged by Mr. David Isherwood.

In publication 7, the author proposed the idea of the application layer packet size optimization and made the theoretical study about the expected packet loss and delay characteristics observed at the application layer when data is streamed over an error prone radio channel. He also prepared and performed the experiments in a WLAN environment to verify the hypothesis resulting from the theoretical study.

In publication 8, the author has performed both the theoretical and experimental analysis of the impact of different parameters, primarily burst length and packet transmission interval, in the bursty transmission mode. He also proposed the idea of using dynamic burst length and implemented the test application to experiment the idea in practice.

Chapter 6 Conclusions

Broadband wireless networks are getting widely available as the commercial deployment of the 3G networks and WLAN hotspots is taking place. Along the new wireless communications networks, the mobile users have access to the most demanding Internet services, such as high quality multimedia streaming, anywhere and any time. However, there are still technical challenges to be solved that are characteristic specifically to wireless networking. Most importantly, radio channel is much more prone to transmission errors than wired links. This is why the traditional streaming applications involving congestion control may not perform optimally in a wireless environment. Another significant problem in mobile telecommunications is the power efficiency. The battery technology is not developing as fast as the processing power, memory capacity and networking technologies.

In this dissertation, some of the most relevant current issues in the multimedia streaming in a wireless environment have been addressed and technical solutions have been proposed. First of all, we have developed a robust framework for streaming audio in AAC format. The concept is based on spreading the individual data elements of each frame, including the separate frame headers and Huffman coded frequency components, among different packets. This arrangement has two major advantages: interleaving allows each frame to be partially reconstructed even if there are some packets missing, and the data elements can be protected separately according to their priority. The advantages have been evaluated via a theoretical analysis, practical experiments and listening tests. The major disadvantage of the scheme is that it modifies the standard AAC bitstream format. Therefore, the proposed system is not compatible with the standard AAC streaming systems. In addition, the interleaving latency restricts the usability of the scheme for highly interactive communications.

In the second part of the dissertation we have studied how the problems caused by bit errors in a wireless link could be mitigated on the application layer of a streaming system. We have performed streaming experiments over a simulated wireless link to show that the proposed streaming strategy employing inter-packet shuffling of data elements provides good robustness against bit errors as well as packet losses. Relaying the packets with bit errors to the application layer could highly improve the utilization of the shared wireless channel and reduce the power usage in the mobile terminal without significant loss in the perceived quality of the streaming multimedia. If the link layer error detection mechanisms do not allow delivery of erroneous packets, an intelligent streaming application may still adapt the packet size at the application layer to reduce the number of link layer retransmissions and improve the power efficiency.

The presented techniques to robust streaming and application layer analysis and adjustment of the streaming parameters could provide substantial improvements in mobile multimedia streaming technologies in terms of the perceived quality, wireless link utilization and power efficiency. The benefits of these techniques are even emphasized when deployed together with some other lately proposed schemes in the field of mobile telecommunications, including the network layer QoS techniques, peer-to-peer streaming, cross-layer optimization with option to relay packets with errors, and power-efficient streaming based on bursty transmission. However, the advantages of the proposed approaches are apparent only when the traffic conditions are not favourable, because of congestion or bit errors in the radio channel. Further research is needed to decide to what extent deployment of these schemes is justified in different realistic telecommunications networks and network conditions.

Most of the proposed schemes are applicable in today's wireless IP networks already. However, the relevance of these proposals will be greatly determined by the evolution of wireless telecommunications technologies and standards in the near future. It is anticipated that the digital broadcast systems, especially DVB-H, will become available in parallel with wireless IP networks. In broadcast and multicast systems, bit error resilience and FEC play a major role due to the lack of a feedback channel. In interactive communications, the important question is the future of the cross-layer optimization in general. Possibility to control the link layer error management mechanisms from the application layer would provide application developers with a powerful tool to get the best out of the adaptive streaming strategies presented in this dissertation.

There are some indicators, most notably the growing interest to UDP Lite and DCCP, suggesting that the option of delivering corrupted packets to higher layers will probably be implemented in several cellular telecommunications systems in the future. The latest research results showing the advantage of cross-layer optimization will most likely encourage development of more advanced cross-layer signaling and control mechanisms as well. In WLAN systems it is not that probable, because the dominating WLAN standard (IEEE 802.11) uses established MAC layer error management mechanisms tightly coupled to the standard. However, it is possible that other technologies for personal area networking, such as Bluetooth, will eventually change the landscape of the wireless local area networking radically. In this case, cross-layer optimization will probably gain more popularity in WLAN communications.

There are several alternative directions to extend the research started in this dissertation. Error management in wireless telecommunications have been studied rather extensively, but more research is still needed to increase our knowledge about the performance of different error management strategies in terms of the subjective quality of the multimedia reproduction when different channel occupancy rates and bit error conditions prevail. Another alternative is to continue the development of robust packetization and interleaving strategies for different multimedia coding paradigms, following the teachings from the experiences with AAC in this dissertation. One possible direction is to bring together the research results of this thesis and the results achieved elsewhere in cross-layer optimization, signaling and bit error management for wireless telecommunications systems. For this option, a concrete step would be to implement an adaptive streaming system, based on the proposed approaches, for a selected wireless telecommunications architecture supporting advanced cross-layer signaling and prioritized transport mechanisms.

References

[1] T. Gold, *Digital Speech Networks*, Proceedings of the IEEE, vol. 65, no. 12, pp. 1636-1658, December 1977.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, IETF RFC 1889, January 1996.

[3] B. Williamson, *Developing IP Multicast Networks, Volume I*, Cisco Press, Indianapolis, Indiana, USA, 1999.

[4] D. Clark, S. Shenker, and L. Zhang, *Supporting Realtime Applications in an Integrated Services Packet Network: Architecture and Mechanisms*, Proc. of ACM Sigcomm, Baltimore, Maryland, pp. 14-26, August 1992.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Service*, IETF RFC 2475, December 1998.

[6] T. Painter, and A. Spanias, *Perceptual Coding of Digital Audio*, Proceedings of the IEEE, vol. 88, no. 4, pp. 451-513, April 2000.

[7] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Järvinen, *The Adaptive Multirate Wideband Speech Codec (AMR-WB)*, IEEE Transactions on Speech and Audio Processing, vol. 10, no. 8, pp. 620-636, November 2002.

[8] M. Dietz, M. Liljeryd, K. Kjörling, and O. Kunz*, Spectral Band Replication, a Novel Approach in Audio Coding*, the 112th Audio Engineering Society Convention, Munich, Germany, May 2002.

[9] R. Schäfer, and T. Sikora, *Digital Video Coding Standards and Their Role in Video Communications*, Proceedings of the IEEE, vol. 83, no. 6, pp. 907-923, June 1995.

[10] T. Sikora, *MPEG Digital Video-Coding Standards*, IEEE Signal Processing Magazine, vol. 14, no. 5, pp. 82-100, September 1997.

[11] T. Wiegand, G. J. Sullivan, G. Bjöntegaard, and A. Luthra, *Overview of the H.264/AVC Video Coding Standard*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, IETF RFC 3550, July 2003.

[13] J. Rosenberg, and H. Schultzrinne, *An RTP Payload Format for Generic Forward Error Correction*, IETF RFC 2733, December 1999.

[14] 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*, IETF RFC 2198, September 1997.

[15] V. Li, and Z. Zhang, *Internet Multicast Routing and Transport Protocols*, Proceedings of the IEEE, vol. 90, no. 3, pp. 360-391, March 2002.

[16] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg, *RTP Retransmission Payload Format*, Internet-Draft, March 2005. Work in Progress.

[17] H. Schulzrinne, A. Rao, and R. Lanphier, *Real-Time Streaming Protocol (RTSP)*, IETF RFC 2326, April 1998.

[18] N. Handley, and V. Jacobson, *SDP: Session Description Protocol*, IETF RFC 2327, April 1998.

[19] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, and G. Fairhurst, *The Leightweight User Datagram Protocol (UDP-Lite)*, IETF RFC 3828, July 2004.

[20] L-A. Larzon, M. Degermark, and S. Pink, *UDP Lite for Real-Time Multimedia Applications*, Proc. of the IEEE International Conference on Communications (ICC '99), Vancouver, Canada, June 1999.

[21] L-A. Larzon, M. Degermark, and S. Pink, *Efficient Use of Wireless Bandwidth for Multimedia Applications*, Proc. of the IEEE International Workshop on Mobile Multimedia Communications (MoMuC '99), San Diego, California, USA, pp. 187-193, November 1999.

[22] R. Sperschneider, D. Homm, and L.-H. Chambat, *Error Resilient Source Coding with Differential Variable-Length Codes and its Application to MPEG Advanced Audio Coding*, the 112th Audio Engineering Society Convention, Munich, Germany, 2002.

[23] R. Sperschneider, D. Homm, and L.-H. Chambat, *Error Resilient Source Coding with Variable-Length Codes and its Application to MPEG Advanced Audio Coding*, the 109th Audio Engineering Society Convention, Los Angeles, California, USA, 2000.

[24] F. Pereira, and T. Ebrahimi (editors), *The MPEG-4 Book*, Prentice Hall PTR, Upper Saddle River, New Jersey, USA, 2002.

[25] C. Dunn, *Efficient Audio Coding with Fine-Grain Scalability*, the 111th AES Audio Engineering Society Convention, New York, USA, September 2001.

[26] R. Geiger, J. Herre, G. Schuller, and T. Sporer, *Fine Grain Perceptual and Lossless Audio Coding based on IntMDCT*, Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03), Hong Kong, vol. 5, pp. V445-V448, April 2003.

[27] S. Johansen, A. Perkins, T. Ramstad, and A. Bopardikar, *Fine-Granular Scalable and Error Resilient Audio Coding by Tree-Structured Quantization*, Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04), Montreal, Canada, pp. 333-336, May 2004.

[28] C. Creusere, *Understanding Perceptual Distortion in MPEG Scalable Audio Coding*, IEEE Transactions on Speech and Audio Processing, vol. 13, no.3, pp. 422-431, May 2005.

[29] V. Goyal, *Multiple Description Coding: Compression Meets the Network*, IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 74-93, September 2001.

[30] C. Aurrecoechea, A. T. Campbell, and L. Hauw, *A Survey of QoS Architectures*, Multimedia Systems, vol. 6, no. 3, pp. 138-151, June 1998.

[31] E. Roberts, *RSVP: A Priority Problem?* Data Communications, vol. 26, no. 7, pp. 58-64, May 1997.

[32] T. Robles, A. Kadelka, H. Velayos, A. Lappeteläinen, A. Kassler, H. Li, D. Mandato, J. Ojala, and B. Wegmann, *QoS Support for an All-IP System beyond 3G*, IEEE Communications Magazine, vol. 39, no. 8, pp. 64-72, August 2001.

[33] S. Maniatis, E. Nikolouzou, and I. Venieris, *QoS Issues in the Converged 3G Wireless and Wired Networks*, IEEE Communications Magazine, vol. 40, no. 8, pp. 44-58, August 2002.

[34] H. Vin, *Multimedia Systems Research: A Retrospective – OR - Whatever Happened to All That QoS Research?* In Proceedings of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '05), Stevenson, Washington, USA, pp. 25-26, June 2005.

[35] A. Ganjam, and H. Zhang, *Internet Multicast Video Delivery*, Proceedings of the IEEE, vol. 93, no. 1, pp. 159-170, January 2005.

[36] W. T. Ooi, *Dagster: Contributor-Aware End-Host Multicast for Media Streaming in Heterogeneous Environment*, Proc. of the Annual Conference on Multimedia Communications and Networking (MMCN '05), San Jose, California, USA, January 2005.

[37] T. E. Starner, *Powerful Change Part 1: Batteries and Possible Alternatives for the Mobile Market*, IEEE Pervasive Computing, vol. 2, no. 4, pp. 86-88, October-December 2003.

[38] E. Nurvitadhi, B. Lee, C. Yu, and M. Kim, *A Comparative Study of Dynamic Voltage Scaling Techniques for Low Power Video Decoding*, Proc. of the International Conference on Embedded Systems and Applications (ESA '03), Las Vegas, Nevada, USA, pp. 292-298, June 2003.

[39] H. Woesner, J-P. Ebert, M. Schlager, and A. Wolisz, *Power-Saving Mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective,* IEEE Personal Communications, vol. 7, no. 2, pp. 34-47, April 2000.

[40] P. Shenoy, and P. Radkov, *Proxy-Assisted Power-Friendly Streaming to Mobile Devices,* in proc. of Multimedia Communications and Networking (MMCN '03), Santa Clara, California, USA, January 2003.

[41] H. Zhu, and G. Cao, *A Power-Aware and QoS-Aware Service Model on Wireless Networks,* Proc. of the IEEE Infocom 2004 (INFOCOM '05), Hong Kong, pp. 1393-1403, March 2004.

[42] S. Chandra, and A. Vahdat, *Application-Specific Network Management for Energy-Aware Streaming of Popular Multimedia Formats*, Proc. of the USENIX Annual Technical Conference (USENIX '02), Monterey, California, USA, pp. 329-342, June 2002.

[43] A. Acquaviva, E. Lattanzi, A. Bogliolo, *Design and Simulation of Power-Aware Scheduling Strategies of Streaming Data in Wireless LANs*, Proc. of the International Workshop on

Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM ’04), Venice, Italy, pp. 39-46, October 2004.

[44] P. Havinga, G. Smit, and M. Bos, *Energy-Efficient Wireless Network Design*, Proc. of the International Symposium on Computers and Communications (ISCC ‘00), Antibes, France, pp. 36-46, July 2000.

[45] P. Havinga, G. Smit, *Energy-Efficient Wireless Networking for Multimedia Applications*, Wireless Communications and Mobile Computing, vol. 1, no. 2, pp. 165-184, April/June 2001.

[46] J. Garcia, and A. Brunstrom, *Checksum-based Loss Differentiation,* Proc. of the IEEE International Workshop on Mobile and Wireless Communications Networks (MWCN ’02), Stockholm, Sweden, pp. 244- 248, September 2002.

[47] M. Welzl, *Passing Corrupt Data Across Network Layers: An Overview of Recent Developments and Issues*, EURASIP Journal on Applied Signal Processing, pp. 242-247, January 2005.

[48] C. Perkins, O. Hodson, and V. Hardman, *A Survey of Packet Loss Recovery Techniques for Streaming Audio*, IEEE Network, vol. 12, no. 5, pp. 40-48, September/October 1998.

[49] Y. Wang, and S. Streich, *A Drumbeat-Pattern Based Error Concealment Method for Music Streaming Applications*, Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’02), Orlando, Florida, USA, pp. 2817-2820, May 2002.

[50] I. Kauppinen, and K. Roth, *Audio Signal Extrapolation – Theory and Applications*, Proc. of the 5th Conference on Digital Audio Effects (DAFX ’02), Hamburg, Germany, pp. 105-110, September 2002.

[51] J. Herre, and E. Eberlein, *Error Concealment in the Spectral Domain*, the 93rd Audio Engineering Society Convention, San Francisco, California, USA, October 1992.

[52] C. Perkins, and O. Hodson, *Options for Repair of Streaming Media*, IETF RFC 2354, June 1998.

[53] G. Liebl, M. Wagner, J. Pandel, and W. Weng, *An RTP Payload Format for Erasure-Resilient Transmission of Progressive Multimedia Streams*, Internet-Draft, October 2004. Work in progress.

[54] J. Nonnenmacher, E. Biersack, and J. Towsley, *Parity-Based Loss Recovery for Reliable Multicast Transmission*, IEEE/ACM Transactions on Networking, vol. 6, no.4, pp. 289-300, August 1998.

[55] N. S. Jayant, and S. W. Christensen, *Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure,* IEEE Transactions on Communications, vol. 29 no. 2, pp. 101-109, February 1981.

[56] M. Yuito, and N. Matsuo, *A New Sample-Interpolation Method for Recovering Missing Speech Samples in Packet Voice Communications*, Proc. of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP ’89), vol. 1, pp. 381-384, May 1989.

[57] N. Iwakami, T. Moriya, and S. Miki, *High-Quality Audio Coding at less than 64 kbit/s by Using Transform Domain Weighted Interleave Quantization (TwinVQ)*, Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’95), Detroit, Michigan, USA, vol. 5, pp. 3095-3098, May 1995.

[58] C. Bisdikian, *An Overview of the Bluetooth Wireless Technology*, IEEE Communications Magazine, vol. 39, no. 12, pp. 86-94, December 2001.

[59] S. V. Saliga, *An introduction to IEEE 802.11 wireless LANs*, Digest of Papers, IEEE Radio Frequency Integrated Circuits Symposium (RFIC ‘00), Boston, Massachusetts, USA, pp. 11-14, June 2000.

[60] L-Å. Larzon, U. Bodin, O. Schelén, *Hints and Notifications*, Proc. of the IEEE Wireless Communications and Networking Conference (WCNC ’02), Orlando, Florida, USA, pp.635-641, March 2002.

[61] Q. Wang, and M-A. Abu-Rgheff, *Cross-layer Signalling for Next-generation Wireless Systems*, Proc. of the IEEE Wireless Communications and Networking Conference (WCNC ’03), New Orleans, Louisiana, USA, vol. 2, pp. 1084-1089, March 2003.

[62] S. Cen, P. C. Cosman, and G. M. Voelker, *End-to-End Differentation of Congestion and Wireless Losses*, IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 703-717, October 2003.

[63] S. Biaz, and N. H. Vaidya, *Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver*, Proc. of the IEEE Symposium on Application-Specific Systems and Software Engineering and Technology (ASSET ’99), Richardson, Texas, USA, pp. 10-17, March 1999.

[64] E. Kohler, M. Handly, and S. Floyd, *Datagram Congestion Control Protocol*, Internet-Draft, March 2005. Work in progress.

[65] P. Lettieri, and M. B. Srivastava, *Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency*, Proc. of the IEEE Infocom (INFOCOM ’98), San Francisco, California, USA, vol. 2, pp. 564-571, March 1998.

[66] E. Modiano, *An Adaptive Algorithm for Optimizing the Packet Size Used in Wireless ARQ Protocols*, Wireless Networks, vol. 5, no. 4, pp. 279-286, August 1999.

[67] J. Yin, X. Wang, and D. P. Agrawal, *Optimal Packet Size in Error-Prone Channel for IEEE 802.11 Distributed Coordination Function*, Proc. of the IEEE Wireless Communications and Networking Conference (WCNC ’04), Atlanta, Georgia, USA, vol. 3, pp. 1654-1659, March 2004.

[68] S. A. Khayam, S. Karande, H. Radha, and D. Loguinov, *Performance Analysis and Modeling of Errors and Losses over 802.11b LANs for High Bit-Rate Real-Time Multimedia*, Signal Processing: Image Communication, vol. 18, no. 7, pp. 575-595, August 2003.

[69] S. S. Karande, and H. Radha, *Does Relay of Corrupted Packets Increase Capacity?* Proc. of the IEEE Wireless Communications and Networking Conference (WCNC ’05), New Orleans, Louisiana, USA, vol. 4, pp. 2182-2187, March 2005.

[70] M. Kornfeld: *DVB-H – The Emerging Standard for Mobile Data Communication*, Proc. of the International Symposium on Consumer Electronics (ISCE '04), Reading, UK, pp. 193-198, September 2004.

[71] S. Kallel, and D. Haccoun, *Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy*, IEEE Transactions on Communications, vol. 39, no. 2, pp. 208-213, February 1991.

[72] H. S. Cheng, G. Fairhurst, and N. Samaraweera, *Efficient Partial Retransmission ARQ Strategy with Error Detection Codes by Feedback Channel*, IEE Proceedings in Communications, vol. 147, no. 5, pp. 263-268, October 2000.

[73] M. Lesk, *Bigger Share of a Smaller Pie*, IEEE Security & Privacy Magazine, vol. 2, no. 4, pp. 62-64, July 2004.

[74] I. Romdhani, M. Kellil, H-Y. Lach, A. Bouabdallah, and H. Bettahar, *IP Mobile Multicast: Challenges and Solutions*, IEEE Communications and Surveys, vol. 6, no. 1, pp. 18-41, 2004.

[75] A-E. Taha, H. Hassanein, and H. Mouftah, *Extensions for Internet QoS paradigms to Mobile IP: A Survey*, IEEE Communications Magazine, vol. 43, no. 5, pp. 132-139, May 2005.

Publications

[P1] J. Korhonen, *Error Robustness Scheme for Perceptually Coded Audio based on Interframe Shuffling of Samples*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02), Orlando, Florida, May 2002, pp. 2817-2820.

[P2] J. Korhonen, *Robust Audio Streaming over Lossy Packet-Switched Networks,* Proceedings of the International Conference on Information Networking (ICOIN '03), Jeju Island, South Korea, February 2003, pp. 1343-1352. Reprinted in ICOIN 2003 Revised Selected Papers, LNCS 2662, Springer-Verlag, Berlin, pp. 386-395, 2003.

[P3] J. Korhonen, and Y. Wang, *Schemes for Error Resilient Streaming of Perceptually Coded Audio*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ‘03), Hong Kong, April 2003, vol. 5, pp. 740-743. Reprinted in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ‘03), Baltimore, Maryland, USA, July 2003, vol. 3, pp. 165-168.

[P4] J. Korhonen, and R. Järvinen, *Packetization Scheme for Streaming High-Quality Audio over Wireless Links*, Proceedings of the Workshop on Multimedia Interactive Protocols and Systems (MIPS '03), LNCS 2899, Naples, Italy, November 2003, pp. 42-53.

[P5] J. Korhonen, *Adaptive Multimedia Streaming for Heterogeneous Networks*, Proceedings of the Conference on Wired/Wireless Internet Communications (WWIC '04), LNCS 2957, Frankfurt Oder, Germany, February 2004, pp. 248-259.

[P6] J. Korhonen, Y. Wang, and D. Isherwood, *Towards Bandwidth Efficient and Error Robust Audio Streaming over Lossy Packet Networks*, Multimedia Systems, vol. 10, no. 5, August 2005, pp. 402-412.

[P7] J. Korhonen, and Y. Wang, *Effect of Packet Size on Loss Rate and Delay in Wireless Links*, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’05), New Orleans, Louisiana, USA, March 2005, pp. 1608-1613.

[P8] J. Korhonen, and Y. Wang, *Power-Efficient Streaming for Mobile Terminals*, Proceedings of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV ‘05), Stevenson, Washington, USA, June 2005, pp. 39-44.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland