Rate-Distortion Optimization for Video Communication in Resource Constrained IP Networks

Stian Johansen

Centre For Quantifiable Quality of Service in
Communication Systems — *Centre of Excellence*
Department of Electronics and Telecommunications
Norwegian University of Science and Technology

A thesis submitted to the
Norwegian University of Science and Technology
Faculty of Informatics, Mathematics and Electrotechnics

for the degree of
Doctor of Philosophy

December 2007

Abstract

Resource limitations lead to a number of challenges in all communication systems. This is also the case for video communication over networks based on the Internet Protocol (IP), where limited rates are shared among competing, heterogeneous users. Packet losses, delays and connectivity losses will be experienced to a varying degree depending on network loads, physical properties and mobility-related issues. These factors, as well as source coding characteristics, influence the visual quality experienced by the users.

One of the main contributions of the work presented in this thesis is that performance gains can be attained when considering characteristics of source coding, networks and congestion control *jointly*. Throughout the presented work, optimization of visual quality is at the centre of attention. The thesis is divided into three main parts, with contributions as follows.

Part A deals with rate-distortion optimized packet loss protection when communicating video over multiple channels simultaneously. Source coder characteristics and characteristics of the (logically or physically) different channels are taken into account in order to yield an optimized packet loss protection of the video. This part presents different optimization algorithms, which are in turn compared in terms of both performance and complexity.

Part B uses the algorithms of part A in the context of congestion control. Specifically, the potential problem of misbehaving receivers is considered. In current systems, there exists an incentive for non-conformant congestion control by video receivers in that an improved video quality can be achieved through obtaining an unfairly high bandwidth share. Since this has unfortunate effects on the connection characteristics of competing users, it poses a potential problem for mass deployment of UDP (User Datagram Protocol) based video services. In this work, a joint source-channel coding based framework which removes the incentive for bandwidth 'greediness' is introduced. Specifically, the framework attempts to reverse the situation and provide an incentive *in terms of visual quality* for adhering to congestion control guidelines for fair bandwidth sharing. The framework is developed for both unicast and multicast cases, and is presented along with optimization algorithms and simulation results.

Part C considers real-time video delivery in mobile ad-hoc networks. As this

type of networks exhibit rather harsh characteristics in terms of throughput, packet losses and mobility-induced route losses, new solutions are required. The approach taken in this work is based on a distributed rate-distortion optimization framework, where multiple sources are used concurrently. The system uses scalable video coding and rateless channel codes in order to allow for uncoordinated sources and distributed optimization. The complete system is implemented in a network simulator, and is shown to exhibit considerable performance gains compared to previous proposed systems.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of PhD at the Norwegian University of Science and Technology (NTNU).

The PhD study has been conducted in the period August 2003 to August 2007. Besides the research work, it has included compulsory courses corresponding to one semester of full-time studies, and one year of teaching assistance and student guidance.

During the study period, I have been hosted and funded by the Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Centre of Excellence. The studies were formally conducted at the Department of Electronics and Telecommunication, Faculty of Information Technology, Mathematics and Electrical Engineering. One year of teaching assistance and related duties was kindly funded by the Department of Electronics and Telecommunications. In addition, from September 2006 to March 2007, I was a visiting researcher at Fraunhofer Institute of Telecommunications, Heinrich Hertz Institute (HHI), Berlin, Germany. This stay was funded by Q2S, NTNU.

Professor Andrew Perkis at NTNU has been the supervisor of this work. In addition, Dr. Thomas Wiegand and Dipl.Ing. Thomas Schierl advised parts of the work during my stay at HHI in Berlin.

It is mentioned that this thesis is a part of a larger research project; namely that of the Q2S centre. As such, it should be read as both a regular PhD thesis but also as an attempt to fulfill a part of the research goals of Q2S.

Stian Johansen
Trondheim, Norway
December 2007

Acknowledgements

Although working towards a PhD may seem like a quite lonely undertaking, this is not the case (at least not always!). I am grateful for the support, help and guidance from a number of people.

First and foremost I would like to thank my supervisor Andrew Perkis for believing in me and my work, as well as his most valuable ability to motivate and promote ambition. I would of course also like to thank the Q2S centre for funding me and granting me some extra time to finish my work. In general, I would like to thank all my colleagues at Q2S for many good times as well as giving me valuable insights into areas of expertise that to a large extent were unknown to me when starting my work at the centre. I would also like to thank the Department of Electronics and Telecommunications and its Signal Processing Group for funding the year of teaching duties.

All co-authors of my (or better, *our*) publications deserve a big thanks. I would particularly like to mention Anna Kim and Thomas Schierl, together with whom I have enjoyed significant research progress.

Everyone that helped me out in the finishing stages of writing this thesis: Yuming Jiang, Anna Kim, Trond Skogstad and Erik Hellerud - thank you. Shpend Mirta at HHI for helping out with preparing the video sequences for the ad hoc streaming work. Also thanks to Odd Inge Hillestad, from whom I got the LaTeX template as well as many good discussions throughout our PhD studies.

Last but not least — a big thanks to my family and my dear Anne Britt. Your continued love and support is deeply appreciated.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Details
AF	Assured Forwarding
AIMD	Additive Increase Multiplicative Decrease
AIPD	Additive Increase, loss-Proportional Decrease
AODV	Ad-hoc On-Demand Distance Vector routing
ARQ	Automatic Retransmission Request
AVC	Advanced Video Coding
BBC	British Broadcasting Corporation
BCH	Bose Chaudhuri Hocquenghem code
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Lenght Coding
CBR	Constant Bit Rate
CCID	Congestion Control ID
CGS	Coarse Grain Scalability
CIF	Common Intermediate Format
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
DCF	Distributed Coordination Function
DCT	Discrete Cosine Transform
DELTA	Distribution of ELigibity To Access
DPCM	Differential Pulse Code Modulation
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
DVB	Digital Video Broadcasting
DYMO	Dynamic On-demand MANET routing
EZW	Embedded Zerotrees of Wavelet transforms
FEC	Forward Error Correction
FGS	Fine Grain Scalability
FLC	Fixed-Length Coding

Abbreviation	Details
FLID-DL	Fair Layered Increase-Decrease with Dynamic Layering
GOP	Group Of Pictures
IBSS	Independent Basic Service Set
IDR	Independent Decoder Refresh
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPTV	Internet Protocol based Television
ISM	Industrial, Scientific and Medical band
ISO	International Organisation for Standardization
JPEG	Joint Photographic Experts Group
JSCC	Joint Source-Channel Coding
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
LAN	Local Area Network
LER	Loss Event Rate
MAD	Mean Absolute Difference
MANET	Mobile Ad hoc Network
MC-EZBC	Motion-Compensated Embedded Zeroblock Coding
MCTF	Motion-Compensated Temporal Filtering
MCN	Multi-Channel Neighborhood
MD	Multiple Description (coding)
MDS	Maximum Distance Separable
MINC	Multicast-based Inference of Network-internal Characteristics
MGS	Medium-Grain Scalability
MPEG	Moving Picture Experts Group
MPR	MultiPoint Relay
MRTT	Multicast Round-Trip Time
MSE	Mean Squared Error
NAL	Network Abstraction Layer
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link-State Routing
PAN	Personal Area Network

Abbreviation	Details
PHB	Per-Hop Behavior
PDA	Personal Data Assistant
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
RD	Rate-Distortion
RDE	Rate-Distortion optimized Equal loss protection
RFC	Request For Comments
RLC	Receiver-driven Layered Congestion control
RLM	Receiver-driven Layered Multicast
RO	Rate Optimal
RO-EXP	Rate Optimal Equal loss Protection
RS	Reed-Solomon
RTP	Real-time Transport Protocol
RTS/CTS	Request To Send / Cleared To Send
RTT	Round-Trip Time
RVLC	Reversible Variable Length Coding
SEQ	SEQential optimization
SIGMA	Secure Internet Group Management Architecture
SNR	Signal-to-Noise Ratio
SQ	Scalar Quantization
SSS	Structured Sub-Search
TCP	Transmission Control Protocol
TFRC	TCP-Friendly Rate Control
UDP	User Datagram Protocol
UXP	Unequal (packet) loss Protection
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VLC	Variable Length Coding
VQ	Vector Quantization
WEBRC	Wave and Equation Based Rate Control
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
3D	Three-Dimensional
3GPP	Third Generation Partnership Project

Chapter 1

Introduction

Recent figures indicate that traffic originating from *www.youtube.com* [1] makes up as much as 10% of all traffic in todays Internet [2]. Add to that the current trend of migration from classical TV broadcasting to IP-based services (e.g. the BBC *iPlayer* [3]), and there should be little room for doubt that there exists a significant interest in video over IP.

This trend, undoubtedly fueled by the increasing penetration of broadband last-mile connections, is likely to continue in coming years. In addition to public interest, the technical solutions are gradually falling into place. More efficient video compression schemes (notably the H.264/Advanced Video Coding (AVC) standard [4]) are commercially available, as well as end user terminals with the required computational power and display capability.

However, the influx of different user terminals, access networks and applications (not to mention potential business models) raises some questions that need answering. Without attempting to make an exhaustive list, one could recognize the following points:

- *Scalability and adaptation.* Handling heterogeneous access, display and decoding capabilities of user terminals is both a difficult and an important task. Parts of the solution to this problem are (and has for some time been) in place. Specifically, the MPEG-21 framework [5] facilitates capability information negotiation between the different communicating entities, and scalable/layered video coding schemes (e.g. MPEG-2 [6], H.263+ [7] and the scalable extension of H.264/AVC [8]). Together, these systems and standards have the potential to allow for flexible transmission and tailoring of video data according to negotiations.

- *Error resilience.* Deploying packet-based video services over lossy and/or congested networks means that the end user is likely to experience packet losses in the received video stream. In these cases, it is crucial that the

source and/or channel coding of the video is done in such a way that decoding and presentation of the video stream is still possible (albeit with degraded visual quality). Approaches for handling this include source-based error resilience techniques (e.g. those included in the H.264/AVC standard [9]) and channel coding based techniques. Important contributions in the latter class of approaches is Priority Encoding Transmission (PET) [10], unequal packet loss protection [11] and the recent "rateless" channel codes [12].

- *Network stability and fairness.* As the volume of video data communicated across the Internet increases, this may have unfortunate implications in terms of stability and fairness. The aforementioned "YouTube" service is based on TCP (Transmission Control Protocol) [13], meaning that this traffic will exhibit a behavior similar to regular web surfing traffic. However, when real-time constraints are considered and services providing higher quality (e.g. IPTV (Internet Protocol-based televison) and multi-way communicative services) are deployed, a move from TCP to UDP [14] protocol is inevitable. Video services running RTP (Real-time Transport Protocol) [15] over UDP have traditionally lacked any congestion control mechanism, a fact that can make these services highly unfair towards other types of traffic. In addition, the congestion control approaches that do exist (e.g. TFRC (TCP-Friendly Rate Control) [16]) are normally implemented at the application layer. This makes them susceptible to various (intended or not intended) non-conformant implementations. This can be a problem with both proprietary and open-source operating systems and media players.

- *Mobility and bandwidth issues.* Being able to use video communication services while mobile is not necessarily a reality now, but is expected to be a significant driver for this type of services. This gives rise to a set of additional challenges, since bandwidth tends to be more scarce in a mobile scenario. Connection losses and delays may also be more significant than in the stationary case. This is particularly challenging in mobile ad hoc networks (MANETs), where bandwidth is a premium and route losses occur frequently. New approaches are needed for providing reliable video communication services in these environments.

Although there are probably as many opinions on the challenges involved in IP-based video communication as there are researchers in this field, the above points give an indication of the problem space. The work presented in this thesis does obviously not attempt to answer all of these challenges, but rather focus on a subset of them. An outline of the thesis is given in the following, with indications of where the contributions of this work belong.

1.1 Outline of the thesis

Chapter 2 provides a review of the large body of work that the presented work is based on. In order to keep this part relatively short, emphasis is placed on the topics and works that have been directly used in the course of this work or are closely related to it.

Following this, the thesis is divided into three parts as follows:

Part A: *Rate-distortion optimized packet loss protection for multichannel video communication.* This part of the thesis, found in chapter 3, extends known approaches for packet loss protection to the multichannel case. Different practical optimization algorithms are discussed and compared in terms of both performance and computational complexity. *The main novelties of this work are the extensions to the multichannel case and the proposed optimization algorithms.*

Part B: *Inciting congestion control through video quality.* Congestion control for UDP-based video is based on the assumption that end user applications perform proper rate control and/or send appropriate congestion-specific feedback to the source. However, there exists an incentive for receivers and applications to violate this assumption. Specifically, a greedy user or application may intentionally (or not intentionally) violate congestion control in order to gain increased visual quality through an unfairly high bandwidth share. This problem has been attempted tackled in literature through protocol-based approaches [17] [18] [19]. In this work, this potential problem is approached in a fundamentally different way. Instead of investigating protocol-based solutions, a joint source-channel coding framework is invoked. *The main novelty consists of integrating source coding, channel coding and congestion control with a specific purpose: Yielding a representation of the video data where there exists an incentive in terms of visual quality for receivers to perform proper congestion control.* This part of the thesis is divided into chapters 4 and 5, investigating the cases of unicast and multicast, respectively. Each chapter provides algorithms for implementation (based on those of chapter 3) and simulation results.

Part C: *Multisource video streaming in mobile ad hoc networks.* As indicated earlier, video transmission in MANETs is demanding because of frequent route losses, limited transmission rates and packet losses. Furthermore, when different users in the system are communicating different video streams, it is a challenge to optimally allocate transmission rates to the involved video streams in order to maximize visual quality. To this end, chapter 6 proposes a distributed multisource video streaming solution for MANETs. *The main novelty is the*

integration of distributed rate-distortion optimization with multi-source streaming based on scalable video coding and rateless channel codes.

1.2 Publications

This thesis is based on work that has been published in a number of refereed conference and journal papers. The papers are listed chronologically in the following. Details of the contributions of the author are given as footnotes to each individual publication.

[I] Stian Johansen, Anna Kim, Bjørnar Libæk and Andrew Perkis — "On the Tradeoff between Complexity and Performance of error protection schemes for embedded codes over parallel packet erasure channels". In *Proceedings of the Norwegian Signal Processing Symposium (NORSIG-05)*, Stavanger, Norway, September 2005.

[II] Stian Johansen and Andrew Perkis — "Unequal error protection for embedded codes over parallel packet erasure channels". In *Proceedings of IEEE Workshop on multimedia signal processing (MMSP-05)*, Shanghai, China, October 2005.

[III] Stian Johansen, Anna Kim and Andrew Perkis — "Quality Incentive Based Congestion Control for Multimedia Communication over IP networks". In *Journal of Zhejiang University (Science A)*, Volume 7, Suppl. 1, pp. 7-12, 2006. Presented at Packet Video Workshop, Hangzhou, China, April 2006.

[IV] Stian Johansen, Anna Kim and Andrew Perkis — "Quality Incentive Assisted Congestion Control for Receiver-Driven Multicast". In *Proceedings of IEEE International Conference on Communications (ICC-07)*, Glasgow, Scotland, June 2007.

[V] Thomas Schierl, Stian Johansen, Cornelius Hellge, Thomas Stockhammer and Thomas Wiegand — "Distributed rate-distortion optimization for rateless coded scalable video in mobile ad hoc networks". In *Proceedings of IEEE International Conference on Image Processing (ICIP-07)*, San Antonio, Texas, USA, September 2007.

[VI] Thomas Schierl, Stian Johansen, Andrew Perkis and Thomas Wiegand — "Rateless Scalable Video Coding for Overlay Multisource Streaming in

[I]-[II]: The author developed the multichannel extensions for the loss protection schemes, implemented software and performed simulations.

[III-IV]: The author contributed in developing the framework and the software implementation.

[V-VI]: The author contributed in the development of the rate-distortion framework, the implementation of simulation software and subsequent testing.

MANETs". Submitted for publication in *Journal of Visual Communication and Image Representation, special issue on Resource-Aware Adaptive Video Streaming*, Elsevier B.V., July 2008.

Publications not directly related to the contents of this thesis:

[VII] Stian Johansen, Andrew Perkis, Tor A. Ramstad and Ajit S. Bopardikar — "Fine-granular scalable and error resilient audio coding by tree-structured quantization". In *Proceedings of IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP-04)*, Montreal, Canada, May 2004.

[VIII] Jan Erik Voldaug, Stian Johansen and Andrew Perkis — "Automatic football video highlights extraction". In *Proceedings of the Norwegian Signal Processing Symposium (NORSIG-05)*, Stavanger, Norway, September 2005.

[IX] Andrew Perkis, Peter Svensson, Odd Inge Hillestad, Stian Johansen, Jijun Zhang, Asbjørn Sæbø and Ola Jetlund — "Multimedia over IP networks". In *Telektronikk*, vol. 102, no. 1, pp. 43-56. Telenor, Norway, 2006

[VII]: This paper summarizes the M.Sc. thesis of the author.

[VIII]: This paper summarizes the M.Sc. thesis of J. E. Voldhaug. The author was assistant supervisor for this work.

[IX]: The author contributed to the literature review that is given in this paper.

Chapter 2

Background

This chapter gives a brief overview of the basic techniques and contributions that are of relevance to the work presented in this thesis. It is attempted to keep this relatively condensed; for details the references should be consulted.

Section 2.1 gives a summary of the main techniques and standards in video coding. Strategies for introducing resilience towards errors and packet losses are given in section 2.2, including source-coding based techniques and channel coding. Section 2.3 is more concerned with video transmission issues, and reviews major contributions in congestion control for continuous media. Finally, section 2.4 discusses the basics of mobile ad hoc networks.

2.1 Video coding systems and standards

Basic concepts and recent developments within the field of video coding, including standardization efforts and systems considerations, are considered in this section. The main emphasis is on standards and techniques that have been used during the course of this work and are used as components in the work presented later in this thesis. It should be noted that the presentation given here is limited to coding of *natural* video. That is, coding of video captured from real-world scenes (thus excluding synthetic video). 3-dimensional and multi-view video coding techniques are not in the scope of this work, and are not treated here.

2.1.1 Video coding fundamentals

Existing video coding systems can roughly be divided into two different classes, namely *hybrid* coders and *subband based* coders, of which examples are given in sections 2.1.2 and 2.1.3, respectively. The basics of these two classes are given in the following.

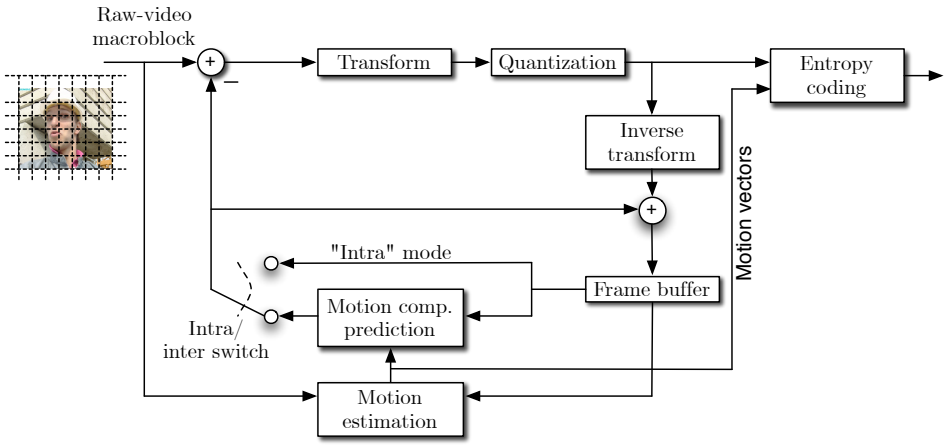


Figure 2.1: Basic block diagram of a generalized hybrid video coder.

Hybrid coding

This class of coders gets its name from the fact that it concurrently uses two different (although related) techniques in order to achieve compression, namely *transform coding* and *prediction*. When visualized using a block diagram, all hybrid coders have a structure similar to that shown in figure 2.1. The structure is similar to that of DPCM (Differential Pulse Coded Modulation), in that the encoded video frames are locally decoded in the encoder, and that this reference (equal to that available in the decoder) is used for prediction of subsequent frames. The basic unit in the encoding stages is known as a *macroblock*, which normally is 16x16 pixels for the luminance (brightness) component, and 8x8 pixels for each of the chrominance (color) components.

The prediction step is known as *motion estimation and compensation*. Here, it is attempted to recreate the current frame (the one that is to be encoded) as closely as possible from one or more already encoded frames. This is done through block-wise translation in the spatial domain. A pixel-based similarity measure (such as MSE, Mean Squared Error, or MAD, Mean Absolute Difference) is computed and used for evaluating candidatae reference blocks. As this is a computationally costly task, the search area for each block is usually limited. When the best (if any acceptable) match is found within this search area, the displacement vector (*motion vector*), is kept, coded and stored/transmitted as side information. The concept of motion estimation is visualized in figure 2.2.

The transform coding step attempts to encode the *difference frame*, the residual image after prediction, as efficiently as possible. This is done through first performing a block-wise transform (traditionally using the DCT, Discrete Cosine Transform), followed by quantization and an entropy coding step. Thus,

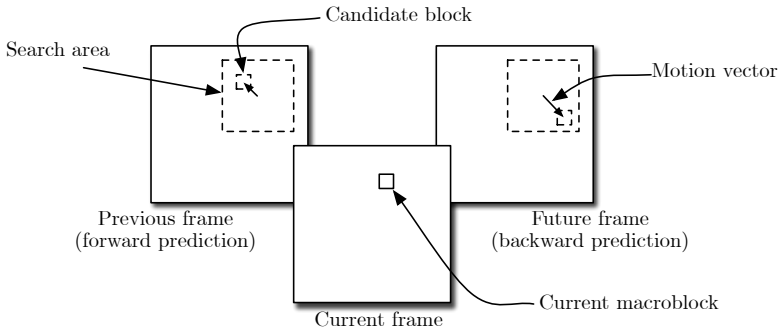


Figure 2.2: Main concepts in block matching for motion prediction.

the transform coding step is structurally similar to that of classical image coders such as JPEG[1] [20]

Subband coding

Known under a few different names (subband coding, interframe wavelet coding, embedded coding, 3D coding), this class of coders separates itself from hybrid coding through extending the filtering/transform operation over the temporal axis as well. Although techniques of this type were proposed some 20 years ago (see [21]), it is not until recent years that they have attracted considerable interest in the research community. Reasons for this are the surge of interest in scalable coding (for which this class of coding schemes are particularly suitable) along with recent developments within the wavelet class of subband techniques.

Using a highpass-lowpass splitting (the simplest being the Haar wavelet [22]) with recursive application on the lowpass channel such as in the subband/wavelet tree decomposition [23] [24], a spatiotemporal decomposition as shown in figure 2.3 is obtained. The resulting subbands can then be quantized directly and entropy coded.

Early approaches within subband coding of video did not use any form of motion compensation, making compression performance inferior to that of its motion compensated hybrid coding counterparts. Approaches using motion compensation have been proposed more recently, known as MCTF (Motion Compensated Temporal Filtering). Here, filtering along the temporal axis is performed along *motion trajectories* that can be estimated for regions/blocks/objects in the video sequence. Using a 2D decomposition for the spatial domain together with MCTF is known as *2D+t* or *t+2D* decompositions, depending on the order

[1]The compression standard is named after the committee which developed it, the Joint Photographic Experts Group (JPEG). This is a joint effort between ISO and ITU-T. The formal name of the JPEG committee is ISO/IEC JTC 1, SC 29, WG 1.

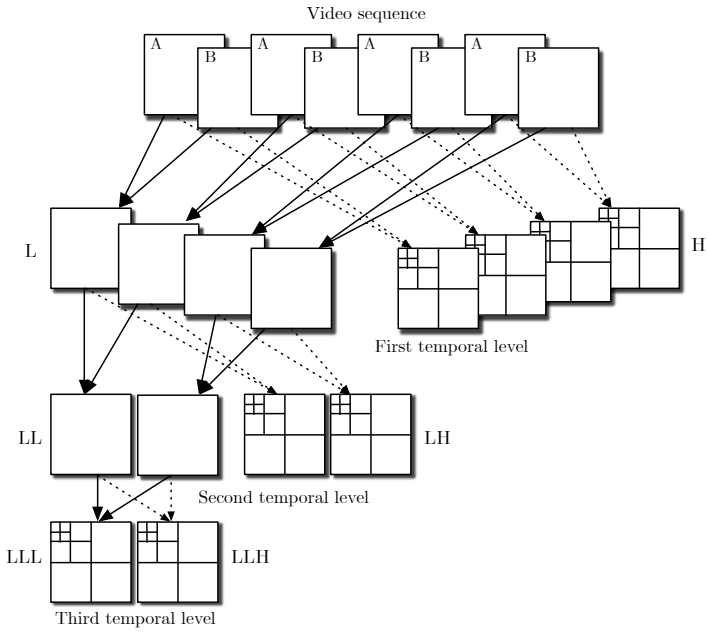


Figure 2.3: Spatiotemporal decomposition with iterative highpass-lowpass (H-L) subband decomposition of video frame pairs. Shown here is a three-level temporal decomposition. A 2-D spatial subband decomposition is done on the nodes in the decomposition tree.

of spatial and temporal filtering. Using MCTF techniques, the performance gap in terms of compression efficiency to hybrid coders has been reduced, although not entirely removed (particularly at low rates).

A desirable property of subband coders is their inherent scalability. SNR (signal-to-noise ratio) scalability can be attained through bitplane coding of subband coefficients (similar to that of the JPEG2000 [25] image coding standard). An example of this way of implementing scalability is given in section 2.1.3. Spatiotemporal scalability can be realized through clever decomposition structures where discarding subbands has the effect of reducing the spatial and/or temporal resolution of the video. For details on this, the reader is referred to [26] and [27], where excellent treatments of the topic are given.

2.1.2 H.264/MPEG-4 Advanced Video Coding(AVC)

H.264/AVC [4] is the latest standardization effort from Joint Video Team (JVT) of ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Motion Picture Experts Group (MPEG). Building upon previous experiences within the field of

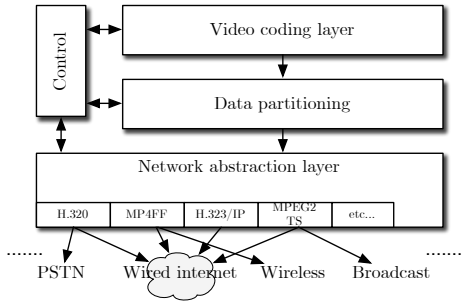


Figure 2.4: Logical interrelation of VCL and NAL in H.264/AVC.

hybrid video coding as well as incorporating more recent advances, H.264/AVC has realized a considerable gain in compression efficiency as compared to previous standards. The standard is suitable for modern services and networks, as it implements a representation and packaging of the video that can be adapted to different services such as broadcast, mobile, video telephony, streaming as well as plain storage. Emphasis has also been placed on error resilience, in order to allow transmission in hostile and error-prone environments.

In order to allow for efficient coding as well as being able to tailor the packaging of data for specific networks and application scenarios, H.264/AVC is based around two architectural concepts: A Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). These two parts are interrelated as shown in figure 2.4 (from [28]). In the following, these two main parts of the H.264/AVC architecture are briefly explained.

Video Coding Layer

The VCL attempts to represent the video content as efficiently as possible. As H.264/AVC is a hybrid coding scheme, the basic structure is similar to what is shown in figure 2.1. Macroblocks are organized to form *slices*, which are the smallest subsets of a frame that can be decoded. Similar to previous standards, H.264/AVC uses the concepts of I (intra coded), P (forward predictive coded) and B (bidirectionally predictive coded) slices. Here, as is normally the case, I slices are encoded without reference to other pictures in the video sequence. Restrictions on the reference picture selection for P and B slices is however relaxed compared to previous standards, replaced with a restriction on memory usage (frame buffering). Two new slice types are introduced, namely *SP* and *SI* slices [29], which are useful for random access, bitstream switching and error resilience. The main feature of SP-frames is that identical SP-frames can be reconstructed by the decoder even though different reference frames are used. The reader is referred to [29] and [30] for detailed treatments of slice types in H.264/AVC.

Transform coding of the prediction residual is performed in a similar fashion to earlier standards, with the exception that different transforms are used. While the 8x8 discrete cosine transform (DCT) is prevalent in other hybrid video coders, H.264/AVC specifies a set of integer transforms of various sizes. Most important is a 4x4 integer transform which has similar properties as the DCT, but avoids inverse-transform mismatch due to the fact that it is specified by exact integer operations (with the added benefit of lowered computational complexity). The smaller transform size typically also reduces "ringing" distortion. However, an 8x8 integer transform is also specified in the so-called "fidelity range extension" of H.264/AVC. This is typically beneficial for high-quality video and high-resolution video, where longer basis functions in the transform are better suited to preserve smoothness and texture details [31] [32].

For entropy coding, H.264/AVC employs two different coding schemes, namely context-adaptive variable length coding (CAVLC) and context-based adaptive binary arithmetic coding (CABAC). The former of these switches between a number of different codebooks (variable length coding (VLC) tables) based on previously coded elements within the same slice. As the name implies, CABAC [33] combines adaptive arithmetic coding with context modeling, providing higher efficiency at the cost of somewhat higher computational complexity.

Finally, it is noted that the inclusion of a content adaptive in-loop de-blocking filter greatly reduces the well-known "blocking" distortion that results from the use of a block transform and independent quantization across different transform blocks.

Network Abstraction Layer

The main task of the NAL is to facilitate the mapping of encoded H.264/AVC data (as resulting from the VCL) to different modes of transport and/or storage. These include RTP/IP [15] over both wired and wireless internet services, ITU-T conversational standards (e.g. the H.32x suite), established broadcasting services including MPEG-2 systems [34] and different file formats.

A *NAL unit* is a packet containing an integer number of bytes, the first byte serving a header purpose for indicating what type of data that is contained in the NAL unit. The NAL units can be of types VCL and non-VCL, depending on whether they are carrying encoded data from video frames or side information such as encoder settings, parameter sets or other information that is essential for successful decoding and playback of the encoded video.

A NAL *access unit* is a collection of NAL units that together can be decoded to yield a single picture. An access unit represents a *primary coded picture*, but may also include *redundant coded pictures* that can be used to recover possible losses and corruption in the parts of the access unit that forms the primary coded picture. An *end of stream* NAL unit is also used to notify the decoder that the last picture of the coded video sequence has been reached.

Parameter sets are used to signal side information that is common to a number of VCL-type NAL units. Two types are defined, *sequence* parameter sets and *picture* parameter sets, being valid for a series of consecutive coded video frames and single video frames, respectively. Since the parameter sets are crucial for successful decoding, they may be repeated if sent in-band with the VCL NAL units, or assigned to more reliable out-of-band transport mechanisms.

Scalable Video Coding (SVC) extension

The H.264/AVC standard as described so far is a non-scalable, single-layer coder. That is, it is not possible to adapt the encoded video bitstream to different user preferences, network characteristics or terminal capabilities by discarding parts of the bitstream. The scalable extension[2] (SVC) [8] [35] of H.264/AVC aims to provide such functionalities by providing a bitstream from which sub-bitstreams with lowered spatial resolution, lowered temporal resolution and/or reduced fidelity (SNR, signal-to-noise ratio) can be easily extracted. Naturally, this is to be achieved with an as small as possible drop in coding efficiency compared to non-scalable H.264/AVC.

Temporal scalability is achieved by carefully selecting which frames that can be used for motion prediction, resulting in a technique known as hierarchical prediction, as shown in figure 2.5 (from [8]). The figure shows a prediction structure that results in four temporal layers, where lower layers can be successfully decoded even though frames belonging to higher temporal layers are discarded. The figure shows a so-called dyadic prediction structure, where frame rates at $\frac{1}{8}$, $\frac{2}{8}$ and $\frac{4}{8}$ of the full temporal resolution are available as sub-bitstreams.

Spatial scalability is achieved by multi-layer encoding where each spatial layer can be independently decoded. As shown in figure 2.6 for the case of two spatial layers, the frames of the video sequence are spatially decimated prior to encoding, with the encoding of each layer employing motion compensation. In order to increase compression efficiency, inter-layer prediction is used [8]. Here, lower layer motion data, prediction residual or reconstructed lower-layer frames are used for improving R-D (Rate-Distortion) performance of enhancement layer encoding.

Fidelity (i.e. SNR) scalability is available in two forms within SVC, namely Coarse Grain fidelity Scalability (CGS) and Medium-Grain fidelity Scalability (MGS)[3]. The former can be seen as a variant of spatial scalability where the

[2]It is noted that the standardization process for SVC is not complete at the time of this writing. The final stages of the standardization process may therefore introduce changes compared to the description given here.

[3]The reason for this naming convention is that another scalability variant, Fine Grain Scalability (FGS) was a part of the standardization process (also, MPEG-4 Visual has an extension for FGS). However, the FGS part of SVC was discontinued due to complexity considerations. The intention of FGS was to allow for fine-grained adaptation of the fidelity and thereby also bitrate.

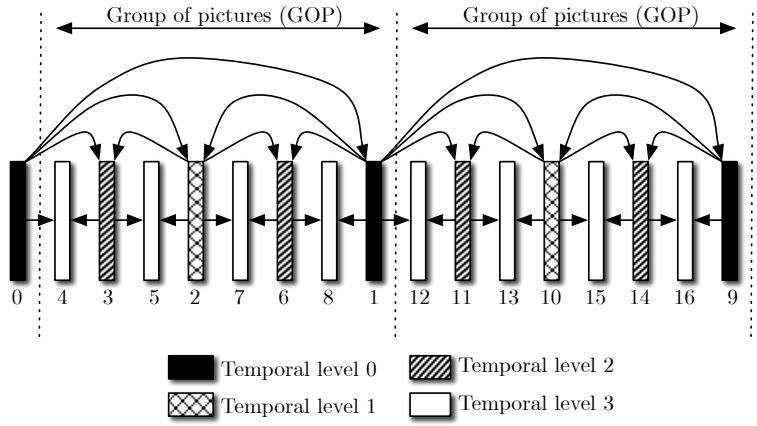


Figure 2.5: Dyadic hierarchical prediction structure for obtaining temporal scalability. The numbers below the frame symbols indicate the encoding order.

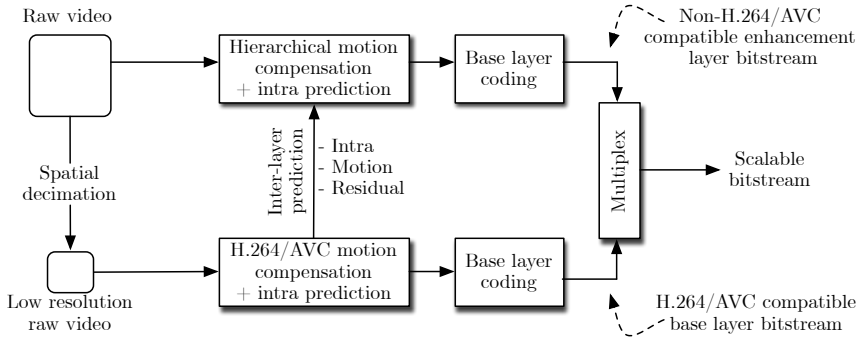


Figure 2.6: Two-layer encoding for spatial scalability in SVC.

decimation and upsampling is bypassed, i.e. the spatial layers have the same resolution. Thus, inter-layer prediction tools successively improve video quality in the enhancement layers through using successively smaller quantization steps. Generating many layers in such a coding structure is however impractical and inefficient, so only a small number of possible fidelities (and thereby bitrates) are realistic. MGS improves on this by allowing for smaller and more closely spaced fidelity layers. This is achieved through a high-level syntax specifying discardable NAL units and a more careful optimization of motion prediction. This involves a trade-off between coding efficiency and effects of the drift problem (which occurs when encoder and decoder have a different reference for motion compensation).

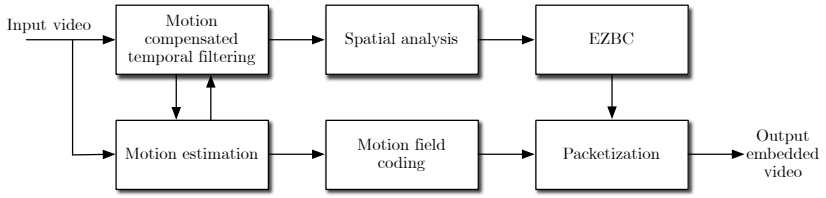
2.1.3 MC-EZBC

In order to avoid the drift problem mentioned, above as well as providing fine grain scalability, *embedded* video coding has been investigated. Building on the seminal paper by Shapiro [36] on embedded image coding using so-called *zerotrees* of wavelet coefficients, the MC-EZBC (motion-compensated embedded zero block coding) technique was developed by Woods et.al. [37] [38]. The basic block diagrams of the MC-EZBC coder and the EZBC subcomponent are shown in figures 2.7(a) and 2.7(b), respectively. After performing motion prediction, MCTF is performed on the video frames along the discovered motion trajectories. A dyadic subband/wavelet decomposition then follows. The EZBC encoding procedure then attempts to create an as efficient representation as possible by exploiting similarities and statistical dependencies across the different bands in the spatiotemporal subband representation. An example of the parent/offspring dependencies in the dyadic decomposition are shown in figure 2.7(c), shown for a 2D-decomposition. The EZBC coding procedure is not explained further here, the reader is referred to [39] and [37] for details.

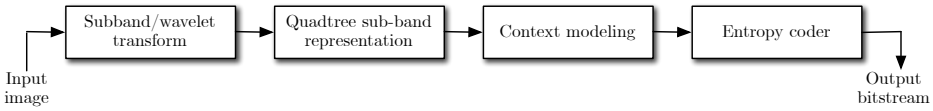
Encoding of the subband values is done by means of context-adaptive arithmetic coding, with subsequent bitplane scanning in order to yield an embedded bitstream. This gives direct fidelity scalability, as the resulting bitstream can be cropped at an arbitrary point, thus providing fine grain SNR scalability. For spatiotemporal scalability, the corresponding subbands in the spatiotemporal decomposition can be dropped, yielding lower temporal and/or spatial resolution.

2.2 Error resilience and packet loss protection

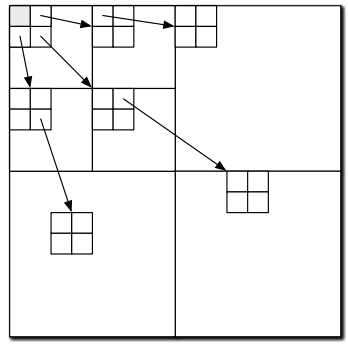
The concept of *error resilience* describes the ability of a system to continue the decoding and display/playout of a media presentation in the presence of impurities during transmission or storage [40]. These impurities may include bit errors (random and/or bursty), packet losses or residual packet losses due to delay. The related concept of *graceful degradation* describes a feature of a communication system where the playout quality gradually decreases with increasing losses/errors. This is in contrast to the well-known "knee effect" where at one point, the decoder can no longer decode at all. This results in a breakdown of visual quality when errors/losses exceed a certain threshold. In the following, some techniques used to provide error resilience and/or graceful degradation are reviewed. The discussion is limited to techniques that are of relevance to the work of this thesis, with an emphasis towards handling of packet losses. Furthermore, error concealment techniques where post-processing is applied after decoding are not reviewed. For a review of such techniques, the reader is referred to [41] and the references therein.



(a) Basic structure of the MC-EZBC video coder.



(b) Block diagram of the EZBC encoding procedure.



(c) Parent/offspring relations in the dyadic subband/wavelet decomposition.

Figure 2.7: The embedded zero block coding (EZBC) coding concepts.

2.2.1 Source coding based error resilience

For introducing error resilience through source coding techniques it is attempted to create a compressed representation of the source material where the effect of potential losses is kept at a minimum. This may include minimizing the spatiotemporal extent/duration of loss-induced inability of decoding, or simply minimizing the penalty in terms of quality degradation as a result of losses.

Entropy coding issues

A common source of decoding failure and quality breakdown is the use of variable length codes (VLC) for entropy coding. *Prefix* codes have the property

that a codeword can never be a prefix of another (longer) codeword, thus ensuring unambiguous table lookup and decoding at the receiver [42]. However, information loss during transmission leads to a loss of synchronization in the decoding process. This synchronization loss makes the decoder (with high probability) unable to decode all data from the point of loss/error to the end of the codestream.

Approaches for minimizing the effects of synchronization loss when using VLCs includes inserting resynchronization markers in the bitstream and using data partitioning. For packet-based communication, it is useful to have a resynchronization marker in the start of each packet, leading to equidistant resynchronization markers in the case of fixed packet size. Data partitioning can be used to further alleviate the effects of losses through separating different types of data into different packets, and applying unequal packet loss protection (discussed in chapter 3) to the different resulting importance levels. Alternatively, priority-enabled network architectures may be used (e.g. DiffServ [43]).

Entropy codes that exhibit better error resilience properties include reversible variable length codes (RVLC) and fixed length codes (FLC). In addition to the prefix condition, RVLCs exhibit the *suffix* condition for codewords. This means that no codeword can be a suffix of another codeword. Thus, as the name implies, these codes can be decoded in both directions, effectively constraining the extent of a loss/error to the lost/corrupted codeword[4]. FLC on the other hand has, obviously, fixed length codewords so that synchronization is never lost. The effect of losses/errors in the FLC case is thus comparable to that of RVLC, depending on the length of codewords. However, both RVLC and FLC incur a penalty in terms of compression efficiency compared to unconstrained VLC. For a review of FLC/RVLC techniques for use in image coding, see [40]. RVLC techniques are suggested for use with MPEG-4 video coding in Annex E of the standard [44] [45].

Multiple description coding

A different means of realizing error resilience and graceful degradation is the family of techniques known as *multiple description* (MD) coding [46] [47] [48]. The aim of MD coding is to encode the source into M "descriptions", where decoding should be possible when receiving any non-empty subset of size m ($1 \leq m \leq M$) of the M generated descriptions. Furthermore, the resulting distortion is (normally) decreasing with the number of received descriptions at the decoder. A key point is that there is no hierarchical structure of the different descriptions, such as the case would be for layered codes or multiresolution structures. The MD coding approach was originally proposed in the late '70s for telephone channel splitting in order to achieve robustness towards link outages, and was further investigated from an information theoretic viewpoint throughout the '80s (see [46]

[4]For multiple errors/losses, the extent of the loss is naturally the codewords between the first error/loss and the last error/loss.

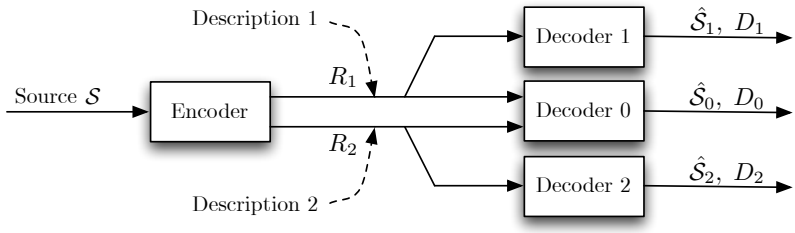


Figure 2.8: Multiple description coding with M=2.

and references therein). It is however not until the past 10 years that applications for MD techniques really have been identified, to a large extent supported by the increasing importance of packet-switched communication. In the extreme case, one could envisage each packet in IP communication being independently decodable, with increasing quality resulting from receiving and decoding further packets (descriptions).

Figure 2.8 shows the simplest case of MD coding, where two descriptions ($M = 2$) of a source $\mathcal{S}$ are generated. The three different decoders in the figure visualize the possibility of decoding either of the descriptions alone or both together. For MD, the following holds for the distortions experienced when decoding:

$$D_0 < \{D_1, D_2\}. \tag{2.1}$$

Here, D_0 is the distortion experienced when decoding both descriptions (central decoder) while D_1 and D_2 are the distortions experienced when decoding the corresponding single descriptions. In general, if the distortions and rates of all single-description decoding cases is equal ($D_i = D_j$ and $R_i = R_j \quad \forall i, j \in [1, \cdots, M]$), we are said to have a *balanced* MD code. It is otherwise considered *unbalanced*.

Making a "good" MD code involves attaining a low distortion when decoding the single-descriptions D_1 and D_2 at their respective rates R_1 and R_2, but still allowing for a good rate-distortion performance of the central decoder (D_0) at the resulting rate $R_0 = R_1 + R_2$. These are somewhat conflicting requirements, and in practice this leads to a performance gap between a non-MD encoded source at the rate $R = \sum_{k=1}^{M} R_k$ and the MD encoded source at the same rate. However, the added error resilience and flexibility of the MD coding approach may justify this in some scenarios.

Several different approaches have been proposed for designing and implementing practical MD coders. A major class of MD coders are MD quantizers, which have been investigated both for the scalar [49] [50] and vector [48] quantization cases. In the seminal paper of Vaishampayan [49], MD coders for $M = 2$ are constructed using two separate quantizers, having the property that they

separately give a reasonable approximation of the input symbol. When combined (equivalent to receiving and decoding both descriptions), the quantization cells intersect in such a way that the resulting distortion is reduced.

Wang et.al. [51] proposed using *correlating transforms* for generating MD codes. The basic idea is, unlike the use of transforms in transform coding, to introduce controlled correlation between the outputs of transforms. If a description is lost, the (known) correlation can be used to better estimate the lost description and thus give a lower resulting distortion.

MD codes can also be implemented using channel codes (forward error correction, FEC). This is closely related to issues discussed in greater detail in section 2.2.2 and chapter 3, so further details on this method for constructing MD codes will be given there.

A very readable overview on the history and theoretical basics of MD coding was given by Goyal in [46], in which the references survey the field from initial idea to more recent advances. A paper discussing topics closely related to parts of this thesis was given by Kim et. al. [52], where MD coding is used in a distributed video streaming scenario in order to provide robustness and for exploiting differences in channel conditions. For video streaming in lossy environments, the comparison by Lee et. al. in [53] gives insights into the performance of MD coding versus layered video coding.

Video coding specifics

While the so far mentioned techniques are more or less general, a number of techniques that are specific to video coding can be used to provide error resilience. A selection of these are surveyed here, with special emphasis on the H.264/AVC standard.

Source coding based error resilience tools in H.264/AVC are found in the VCL layer. These include (but are not limited to) the use of slices, flexible macroblock ordering (FMO), intra macroblocks and reference frame selection [9].

FMO is a tool that is used for dispersing the effects of losses or decoding errors over larger areas in video frames, thereby reducing the resulting perceptual degradation [54]. It also improves the performance of error concealment schemes, since losses of multiple smaller, separated blocks is more easily concealed than larger lost areas (due to the common property of smoothness across block boundaries). Two possible ways of doing FMO are shown in figure 2.9. The two colorings represent two different *slice groups*. The key point is that macroblocks (small blocks in the figure) can be assigned to different slice groups independent of their spatial location in the frame. The improvement in error resilience from FMO does however reduce compression efficiency somewhat, since block assignment is done prior to the encoding procedure, and thereby reducing spatial correlation of the resulting slice groups.

When feedback between the decoder and encoder is available, reference

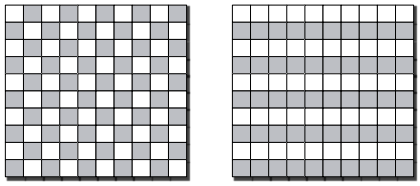


Figure 2.9: Flexible macroblock ordering (FMO). Left: Checkerboard assignment pattern. Right: Interleaving mode.

frame selection can be done based on knowledge of decoding state. In the ACK (acknowledgment) mode, only frames or parts of frames that have been acknowledged by the decoder can be used as reference in inter-frame coding [54]. The reduction in (temporal) error propagation because of this is obvious, but it may also reduce compression efficiency significantly when the delay between encoding a frame and reception of the corresponding ACK is large. In the NACK mode (negative ACK, the decoder only signals inability of decoding frames), the encoder is restricted to using frames that were encoded prior to the frame for which a NACK was received as reference frames. It is also possible to use a combination of the ACK and NACK approaches, in which the encoder switches between them based on what is received from the decoder [54].

An analysis of the efficiency of different error resilience tools in H.264/AVC for streaming of high definition content can be found in [55]. For a more complete overview of error resilience tools in H.264/AVC and their usability for IP-based communication, see [9].

2.2.2 Channel coding

Channel coding techniques for the correction of errors and losses in communication certainly is a vast field, comprising work from its infancy in the 1940's up to modern techniques that are continually being invented, improved and applied to new scenarios. The range of different channel coding techniques and their application areas is indeed large, so any attempt at giving a complete review in this context is not feasible. Rather, the short introduction given here limits itself to the classes of codes that are used or assumed used in the work of this thesis. These are the *maximum distance separable* (MDS) codes, of which the Reed-Solomon code is an example, and the more recent *rateless* codes, of which the so-called Raptor code is described here. For general treatments of channel coding techniques, the reader is referred to e.g. [56] and [57].

MDS codes and the Reed-Solomon code

The Reed-Solomon (RS) code [58] is a cyclic error correcting code which can be considered a subclass of the BCH (Bose-Chaudhuri-Hocquenghem) codes [56]. Cyclic codes are in turn a subclass of linear block codes, characterized by the parameters n and k, where n is the block size (codeword length) and k is the number of information symbols per codeword. Thus, $n - k$ redundancy symbols are added in the process of encoding. RS codes furthermore belong to the class of codes that are *maximum distance separable* (MDS). This means that the RS code (as all MDS codes) achieves the *Singleton* bound

$$d_{\text{free}} \leq n - k + 1 \tag{2.2}$$

with equality. Here, d_{free} represents the *free distance* or *minimum distance* of the code, which is the smallest possible number of linearly dependent columns of the parity check matrix (H) that is associated with the code $\mathcal{C}$. A more practical interpretation of d_{free} is the lowest number of places (digits) in which two codewords can differ.

The RS code is able to correct up to t errors, as given by

$$t = \left\lfloor \frac{d_{\text{free}} - 1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor. \tag{2.3}$$

A practical interpretation of this is that, in order to correct an error, the decoder needs one redundant symbol to identify the error and another to correct it [59]. This is valid when the location of errors in the received symbol stream is unknown. A different transmission failure case is *erasures* (e.g. packet losses), where the location of lost data is known although no symbols contained in this erasure have been received. The erasure-correcting capability e of the RS code is twice the error correcting capability:

$$e = d_{\text{free}} - 1 = n - k. \tag{2.4}$$

Note that, for RS codes, this holds regardless of the location of erasures (i.e. whether erasures happen in the redundancy or the source digits).

Rateless codes

As described in the previous section, block codes are characterized by the parameters k and n, yielding a *code rate* k/n. Naturally, these are design parameters that have to be decided prior to encoding based on knowledge of the channel/application at hand. Recently, a new class of error correcting codes has been developed, known as *rateless* codes (also known as *fountain* codes) that are fundamentally different in this respect. These codes have the property that the number of encoding symbols that can be generated from the source symbols

is theoretically unlimited. Thus, the notion of a code rate does not exist for these codes (hence their name). Consequentially, the "strength" of the code (as given by the code rate in the case of block codes) does not have to be decided prior to encoding. This is certainly useful if the channel conditions or application is not known at the time of encoding. Furthermore, the code family also has some elegant properties in combination with packet based communication and in multi-source scenarios, as will be utilized in chapter 6 in this thesis.

In general, a rateless code is able to reconstruct the k source symbols using *any* possible set of k' received (encoded) symbols. An optimal code will have $k' = k$, meaning that there is actually no overhead introduced. Practical codes (as discussed in the following) should, with high probability, be able to decode the k source symbols with $k' \geq k(1 + \epsilon)$, where ϵ characterizes the efficiency of the code and its implementation.

The first practical rateless code was the *LT code* of Luby [60]. A version of the LT code was then used as part of the later work by Shokrollahi [61], where the *Raptor* code is introduced. The Raptor code has the attractive property of computationally efficient (linear time) encoding and decoding, while allowing for a small ϵ for most interesting input sizes k. The theoretical basis and details of encoding and decoding of Raptor codes is not restated here, the reader is referred to [61] and [12] for good treatments of this. Here, it suffices to say that the Raptor code is a concatenated code, constructed using the LT code as the inner code and a regular block code (e.g. an LDPC, low density parity check code) as the outer code. Practical implementations achieve an overhead ϵ approximately in the range 0.03 to 0.1, depending on the size of source data and its statistics.

MD-UXP: Channel coding based MD coding

Multiple description systems can be constructed using channel codes. This is conceptually different from the source coding based approached discussed earlier. This approach is closely related to the Priority Encoding Transmission (PET) approach of Albanese et. al. [10], and was further developed and identified as a variant of MD coding by Mohr et. al [11]. The concept is known as multiple description unequal loss protection (MD-UXP or MD-FEC), and the main idea is shown in figure 2.10.

An embedded bitstream (such as e.g. the EZW image coder or the MC-EZBC video coder discussed earlier) is considered as source data. The bitstream is divided into equal-sized blocks, as depicted in figure 2.10 a). By assigning error correction codes (denoted FEC, forward error correction, in the figure) of different strengths to different parts of the bitstream, an MD-like representation is generated. Consider that each description (i.e. packet) is found as a column in the matrix-like representation of source data and redundancy in the figure. This means that each description contains a *part* of each codeword/codeblock that is found as the rows in this representation. A closer study of figure 2.10 b) now

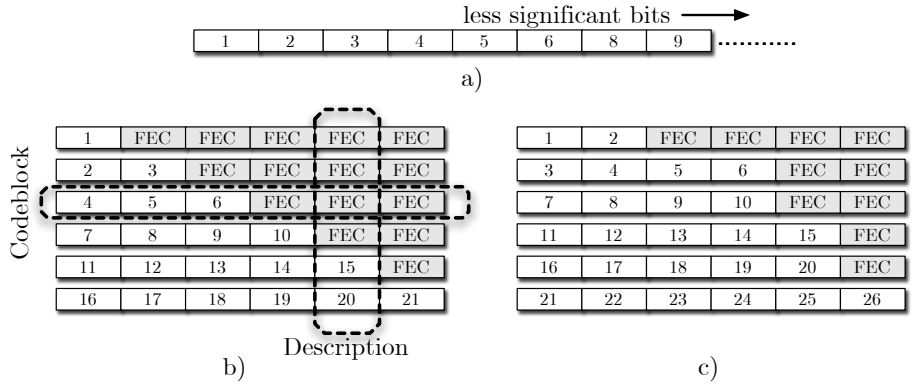


Figure 2.10: a): Embedded, encoded bitstream. b) and c): Two different configurations of MD-UXP.

reveals the following. Consider that the erasure correcting code used has the MDS property, and that it furthermore is oblivious to the location of erasures within each codeword[5]. Reception of *any* one description would allow for block 1 of the the embedded bitstream to be successfully decoded, reception of two descriptions would allow for another two source blocks to be decoded, and so on. Finally, reception of all six descriptions would allow for decoding of the bitstream using all 21 source blocks. Thus, the MD property that each description should be equally useful for reconstructing data, and that reception of more descriptions should enable a better (i.e. higher quality) reconstruction, is fulfilled.

However, such an assignment of error correcting codes as in figure 2.10 b) may be wasteful if channel conditions do not justify such strong erasure protection. This realization lead to the development of channel state optimized and rate-distortion optimized versions of the above outlined scheme, with an example allocation of error correcting codes shown in figure 2.10 c). It should however be noted that schemes like that of figure 2.10 c) can not strictly be said to be an MD representation, since the reception of only a single description does not allow decoding at all, neither does the distortion experienced at the decoder decrease monotonously with the number of received descriptions. MD-UXP techniques are related to the topic of chapter 3, so further discussion is found there.

2.3 Congestion control in video communications

In shared communication networks like the public internet, congestion control is a necessity in order to maintain reliable communication with predictable delays.

[5]An example of such a code is the Reed-Solomon code as discussed earlier.

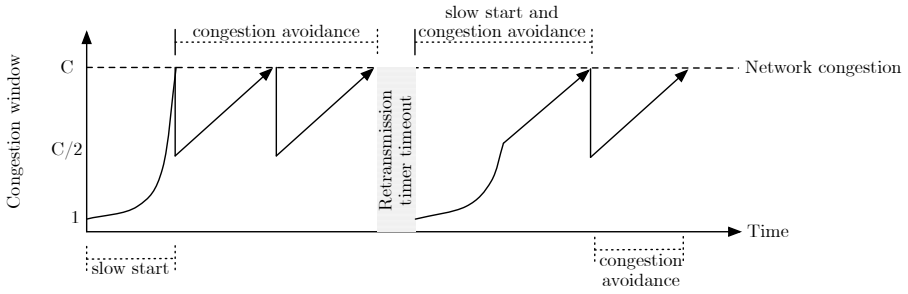


Figure 2.11: Conceptual variation of the congestion window in TCP congestion control.

In the context of continuous-media communication services (like video streaming and conversational video), congestion control strategies should ideally be tailored for this application, because of its differing requirements compared to reliable services like file transfer, web browsing and the like. In this section the most important congestion control schemes are summarized, with emphasis towards those of relevance to video services in general and the work of this thesis. The unicast case is considered first, followed by a treatment of the multicast case.

2.3.1 Unicast

Unicast, where the packets of a stream have a single source and a single destination, is by far the most prevalent routing scheme. The dominant TCP (Transmission Control Protocol) [13] protocol has its built-in window-based congestion control mechanism of the increase-decrease type. Roughly, the TCP congestion control can be divided into two different modes of operation, *slow start* and *congestion avoidance*. In the slow start phase, which is used at the startup of a new TCP connection or after restart due to the expiration of a retransmission timer, an exponential growth approach is used. The source sends one TCP *segment* (i.e. a packet), waits for an acknowledgment (ACK) from the receiver, then transmits two segments, then four after receiving a new ACK, and so on. This is done until congestion is reached (which is detected by a packet loss due to router buffer overflow), at which point the *congestion window* is reduced to half, and the congestion avoidance phase is entered. In this phase, the congestion window is increased linearly by at most one segment per RTT (round-trip time) until congestion is reached and the congestion window is halved again. A sketch of how the congestion window may vary with time is given in figure 2.11.

TCP is a reliable protocol where lost packets are retransmitted until an ACK for the corresponding segment is received at the source. Certainly, this will lead to significant delivery delays in cases where congestion or other sources of packet

loss (e.g. wireless links) is an issue. This, combined with the highly fluctuating "sawtooth" transmission rate behavior of TCP leads to the conclusion that it is not well suited for continuous media. For these types of services packet losses can to an extent be tolerated but significant delays can not, and the allowable transmission rates should vary relatively smoothly with time. For these reasons, the unreliable and connectionless UDP (User Datagram Protocol) [14] protocol with standalone additional congestion control (such as TFRC, as described in the following) is normally used for continuous-media communications.

TFRC

TCP-friendly rate control (TFRC) [16] [62] has been proposed by Floyd et. al. for addressing the main shortcomings of TCP in the context of continuous-media communication. It is a congestion control mechanism that does not specify its own protocol, but rather implements feedback-based control of the sending rate at the application layer. It is designed to be reasonably fair towards competing TCP traffic, meaning that the transmission rate of a TFRC session should be within a factor of two of the TCP rate under the same conditions[6]. TFRC is an *equation based* approach, meaning that the appropriate sending rate is calculated based on a relation involving the relevant parameters. Specifically, TFRC uses an equation that approximates the behavior of the Reno variant of TCP as follows [63]:

$$X = \frac{s}{RTT\sqrt{\frac{2bp_{\text{ler}}}{3}} + (t_{\text{RTO}}(3\sqrt{\frac{3bp_{\text{ler}}}{8}})p_{\text{ler}}(1+32p_{\text{ler}}^2))}, \tag{2.5}$$

where X is the transmission rate (bytes/sec), s is the segment (packet payload) size in bytes, t_{RTO} is the TCP retransmission timeout value in seconds, b is the number of packets that are acknowledged by a single ACK and p_{ler} is the *loss event rate*. Note that the loss event rate is different from the more common measure of loss fraction (packet loss rate), in that one or more packet losses within a single RTT is counted as a single loss event. The loss event rate is calculated as the inverse of the average loss interval, which is found using a weighted moving average. The averaging ensures that the parameter p changes smoothly, thus giving a reasonably smooth variation of the rate X with time. The weights in the averaging filter may be tuned to adjust the rate smoothness, but a "too smooth" X will compromise fairness towards competing TCP-based traffic.

AIPD and AIMD

As we have seen, in its congestion avoidance phase TCP employs a strategy where the sending rate is normally halved upon detection of a packet loss and

[6]Having a sending rate within a factor of two of TCP under the same conditions is normally referred to as being *TCP friendly*. A slightly different (and more restrictive) concept is *TCP compatibility*, in which a flow uses no more bandwidth than TCP under the same conditions [62].

is otherwise increased linearly. This behavior belongs to the AIMD (Additive Increase, Multiplicative Decrease) family of congestion control schemes, which are characterized by the relations

$$X_{i+1} \leftarrow X_i + \alpha \ \text{(increase)} \tag{2.6}$$

$$X_{i+1} \leftarrow X_i(1-\beta) \ \text{(decrease)}. \tag{2.7}$$

Here, β would be 0.5 for the rate-halving variant of TCP. A slightly different approach that can potentially incur more dampened rate fluctuations is AIPD (Additive Increase, loss-Proportional Decrease), which is characterized by

$$X_{i+1} \leftarrow X_i + \alpha \ \text{(increase)} \tag{2.8}$$

$$X_{i+1} \leftarrow X_i(1-\gamma p) \ \text{(decrease)}. \tag{2.9}$$

The decrease parameters p and γ are the packet loss fraction and the proportionality constant, respectively. A good comparison of AIMD and AIDP is given in [64], where it is concluded that AIPD competes more aggressively for bandwidth than AIMD, and may not be TCP friendly. However, depending on the settings of the α and γ parameters, a lower fluctuation of the rate X than in the AIPD/TCP case can be achieved.

DCCP

Although not a congestion control scheme in itself, DCCP (Datagram Congestion Control Protocol) [65] [66] deserves to be mentioned in this context. DCCP is a recent proposed IETF (Internet Engineering Task Force) specification. The design aim of DCCP was to develop a lightweight congestion-controlled protocol for unicast communication, without the reliability feature of TCP. The lack of a (mandatory) reliability feature makes it, like UDP [14], suitable for delay-constrained continuous media. Applications may choose from a set of different standardized congestion control algorithms via Congestion Control IDs (CCIDs), depending on the requirements of the application and the scenario in which they are to be deployed. Currently, DCCP is limited to providing *TCP-like congestion control* (CCID 2) [67] and TFRC (CCID 3) [68].

2.3.2 IP Multicast

For scalability towards larger populations of receivers, multicast may be preferred over unicast. In multicast, receivers subscribe to multicast *groups* through the Internet Group Management Protocol (IGMP) [69]. Roughly speaking, sources send packets to a designated multicast address and routers then forward packets to those receivers/routers that have subscribed to the relevant group. For scalability reasons, feedback from receivers to the source is avoided in multicast. This, along with the fact that different receivers of the same session may be on

different networks with different capacities and loads, makes congestion control for multicast very different from the unicast case.

Specifically, *receiver driven* congestion control and rate adaptation is normally used in the multicast case. This distributed congestion control approach allows for adaptation to local congestion experienced by the individual receivers in the multicast session. In the following, the most important receiver-driven multicast schemes with applications in video communications are briefly reviewed.

Receiver-driven layered multicast (RLM)

The first of its class, RLM was proposed by McCanne et. al. in 1996 [70]. It is based on using a layered video codec and distributing the individual layers through separate multicast groups. The basic logic is that receivers drop/unsubscribe a layer upon detection of congestion and add/subscribe a layer when there is spare capacity.

A potential problem arises when a receiver attempts to add a new layer to its subscription ("join experiment"), potentially forcing other users to drop a layer if the experiment fails (i.e. congestion occurs as a result). The employed strategy for avoiding these losses is a shared learning approach, where different *join timers* are used for the different layers. The join timer for a layer is multiplicatively increased if the join experiment fails, resulting in more rare attempts for join experiments that are more likely to fail. Furthermore, receivers multicast a message to other receivers when it attempts to add a layer. Through this approach, receivers can learn from the success/failure of other receivers' join experiments and use this information for adjusting their subscription levels.

Receiver driven layered congestion control (RLC)

Proposed by Vicisano et. al. [71], RLC differs from RLM in that it attempts to address the problem of fairness towards TCP. The potentially problematic RLM approach of inter-receiver notifications during join experiments is also avoided through the use of *synchronization points* in the stream(s). Receivers are only allowed to increase subscription levels at or immediately after reception of a synchronization point, but may reduce subscription levels at any time (upon detection of congestion). The source also inserts *probing bursts* into the network, in order to emulate subscription of an additional layer. Receivers suppress reactions to congestion during this probing period. A lack of congestion during the probing burst is an indication that another layer may be added. The main advantage of the probing approach is that it reduces the frequency of join experiments, which have relatively long-lasting effects if they fail. The reason for the sustained effect of failed join attempts is the IGMP leave latency, which can be (depending on router configuration) up to 9 seconds [72]. However, the probing approach has a shortcoming in that the burst length may be too short to result in congestion, leading to erroneous join decisions by receivers.

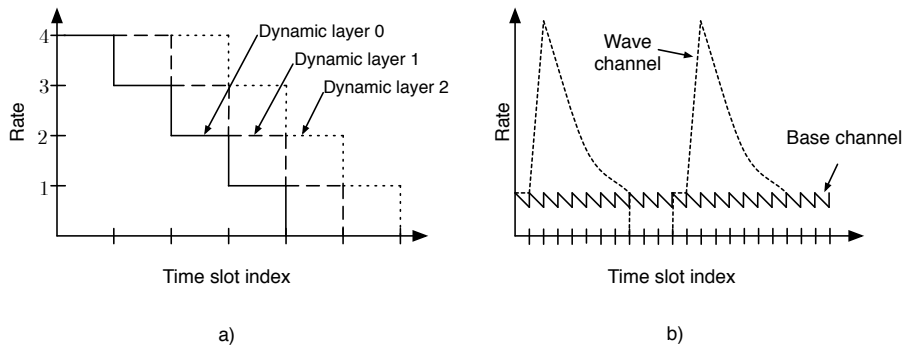


Figure 2.12: Dynamic layering for FLID (left) and WEBRC (right).

Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL)

FLID-DL [72] was proposed by Byers et. al. as a generalization of RLC, especially targeting the latency problem associated with IGMP when leaving multicast groups (i.e. reducing the subscription level). FLID is based on the use of a rateless code (see section 2.2.2), which allows for some flexibility compared to the non channel coded case with respect to packet scheduling.

The concept of dynamic layering implies that the rate of each layer *decreases* over time, having the effect that if a receiver wishes to maintain its sending rate it must subscribe to "new" layers at regular intervals. This is illustrated in figure 2.12 a). Refraining from joining new layers thus equates to reducing the reception rate. This effectively suppresses the effects of the IGMP leave latency, yielding a faster response to congestion[7]. A notable drawback of the FLID-DL scheme is that the frequency of IGMP join/leave messages can become unwieldy, possibly hindering its deployment in practice.

Similar to RLC, FLID employs synchronization points (known as *increase signals*) to indicate when receivers may increase their subscription levels. The placement and frequency of these signals is derived depending on the rates of different layers in order to emulate the congestion response characteristics of TCP, as approximated by equation 2.5.

Wave and Equation Based Rate Control (WEBRC)

The so far described multicast multirate congestion control schemes all belong to the increase/decrease class. WEBRC [73], on the other hand, is *equation based*. Similar to TFRC, WEBRC directly uses an analytical approximation of TCP throughput (see equation 2.5) for calculating the fair rate. RLC and FLID also attempt to shape their rates according to TCP, but it is in those cases done

[7]Note that the FLID algorithm may be used with static layering (SL) as well, at the cost of a slower response to congestion but with the advantage of less frequent join messages.

implicitly through the placement of synchronization points, choice of layer rates and increase/decrease strategies.

Similar to FLID-DL, WEBRC uses highly time-varying sending rates for the different multicast groups and is implemented using rateless codes. Specifically, the rates of each multicast group are periodic and exponentially decreasing from a recurring maximum (hence the "wave" name), as illustrated in figure 2.12 b). A small mandatory base channel with almost fixed rate is subscribed by all receivers, and further rate is subscribed by joining the wave channels at some point of their descent (see figure 2.12). Joining the waves close to their maximum rate yields a high accumulated rate, while joining lower on the curve will correspondingly yield a lower accumulated rate.

A difficulty with using equation 2.5 in the multicast case is that the RTT is ill-defined due to the lack of closed-loop communication between source and receiver, as well as the heterogeneity of receivers. To this end, WEBRC is based on a multicast-centric variant of RTT, known as MRTT. The MRTT is found as the time between when a "join" message is sent for a wave and the time at which the first packet from the wave is received. Further details on the calculation of MRTT can be found in the original paper [73].

Finally it is noted that WEBRC is mainly intended for reliable data delivery, although the authors state that it may be applied for video streaming purposes. The applicability of WEBRC in practice for such services, however, is not well investigated.

2.4 Mobile Ad Hoc Networks (MANETs)

A MANET [74] can be defined as an autonomous system of mobile hosts connected through wireless links, in which all hosts may serve as routers (see figure 2.13). A MANET does not rely on existing infrastructure nor centralized administration, a property that makes it interesting in certain scenarios. The notion of MANETs can be traced back to the work on the ALOHA protocol in 1968 [75] [76], but has recently been subject to an influx of research and also a certain commercial interest. MANETs have obvious applications within defense/battlefield communications and in disaster/emergency communication scenarios where existing infrastructure may not be available or has broken down. More recent advances in shorter-range wireless technologies such as the IEEE[8] family of Wireless LAN (WLAN, Wireless Local Area Network) [77] standards and Personal Area Network (PAN) standards such as Bluetooth [78] have led the way for deployment of MANETs in everyday communication scenarios. MANETs can be used for extending the range of fixed infrastructure, allowing network access to out-of-range hosts and/or hosts without fixed access. Another important

[8]Institute of Electrical and Electronics Engineers Inc.

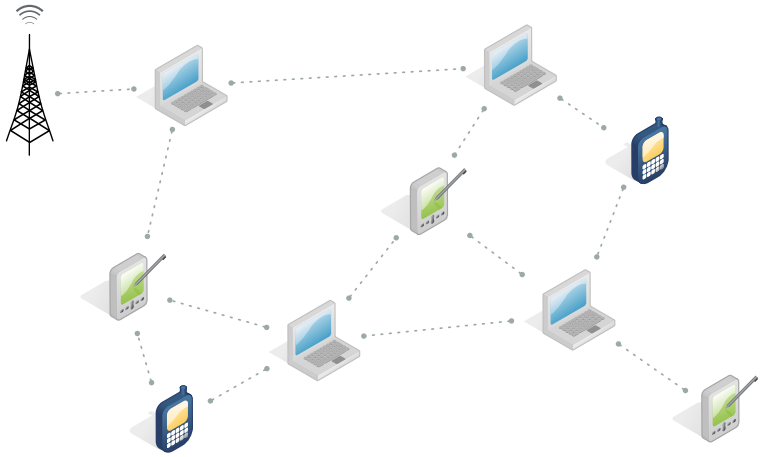


Figure 2.13: Example MANET scenario. Laptops, PDAs (Personal Data Assistants) and mobile phones share the network connection of a single laptop through a multihop MANET.

application area is low-power sensor networks that can be used for monitoring and surveillance.

2.4.1 Routing

Since all nodes in a MANET may act as routers, direct application of well-known routing strategies from the wired internet is neither possible nor efficient. The mobility of nodes leads to highly time-varying routing tables and the need for nodes to continuously discover and re-discover neighboring nodes. MANET routing strategies can roughly be divided into two categories, *reactive* and *proactive*. These strategies are briefly reviewed below.

Reactive MANET routing

Also known as *on-demand*, reactive routing sets up routes only when connections are requested. An example protocol is Ad-hoc On-demand Distance Vector (AODV) [79]. In AODV, routes are established by broadcasting a route request (RREQ) message. If a node receiving the RREQ is the intended receiver or knows a "fresh" route to the intended receiver, it answers using a route reply (RREP) message using unicast communication. Otherwise, the RREQ is re-broadcasted to nodes within its transmission range. When a route is set up, it will remain active for as long as packets are traversing it. After a timeout value, the route is deleted from local routing tables at nodes. A slightly different approach is Dynamic Source Routing (DSR) [80], which mainly differs from AODV in that

it does not directly use local routing tables at intermediate nodes, but rather makes each source maintain a complete source-to-receiver routing table. Multiple alternative routes may be set up for use if one breaks, in order to avoid the delay needed for repairing the route or finding a new one.

A disadvantage of reactive approaches is the delay associated with establishing routes. The broadcast/flooding strategy used for finding routes may lead to congestion if multiple nodes are requesting route establishment concurrently. On the other hand, the steady-state messaging overhead in reactive protocols is small, since little or no continuous updating of routes is done.

Proactive MANET routing

In proactive (or *table-driven*) routing algorithms, all nodes maintain a routing table to all other nodes in the network at all times. For handling mobility, routing tables may be updated at regular intervals and/or updated based on mobility-related events. An example of a proactive routing algorithm is Optimized Link State Routing (OLSR) [81]. Link state routing is based on making individual nodes calculate the "best" next-hop route to all other nodes based on a graph representing the connectivity in the network. OLSR uses so-called multipoint relays (MPRs) to reduce the messaging load on the network. Control messages from a node are only re-broadcasted from the MPRs that belong to the node in question. Moreover, a node only exchanges link state information with nodes that have selected it as an MPR.

Proactive approaches have the advantage that routes are readily available when needed, so there is little or no delay associated with the setup of communication. On the other hand, steady-state message exchange, processing power and memory requirements may be significantly higher than in the reactive case.

2.4.2 Enablers and technologies for practical MANETs

Off-the-shelf technologies and products that enable deployment of relatively simple small-scale MANETs are becoming available. These include the IEEE 802.11 family of standards and the Bluetooth specification. While Bluetooth has the advantage of energy efficiency and miniaturization, achievable transmission rates are lower than the requirements of many video applications. The discussion here is thus limited to the IEEE 802.11 protocol stack.

IEEE 802.11 a/b/g

Ubiquitous in personal computers, PDAs and other mobile units, the IEEE 802.11 protocol stack dominates the market for indoor WLAN. The IEEE 802.11 working group was formed in 1990, with the goal of defining standard physical (PHY) and medium access control (MAC) layers for the ISM (industrial, scientific

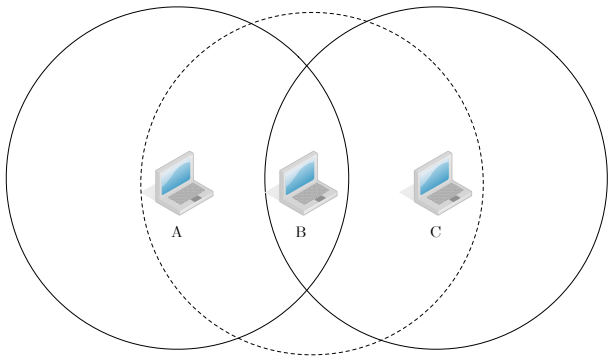


Figure 2.14: Situation with possible "hidden node" problem. Circles indicate the transmission ranges of each node.

and medical) radio bands. The most widely deployed variants 802.11a and 802.11b are residing in the 5.2GHz and 2.4GHz bands, respectively. 802.11b is based on DSSS (Direct Sequence Spread Spectrum), and supports physical-layer bitrates of 11, 5.5, 2 and 1 Mbps. 802.11a utilizes OFDM (Orthogonal Frequency Division Multiplexing) and supports physical layer bitrates up to 54 Mbps. The more recent 802.11g specification attempts to combine the best of the a and b specifications, providing a physical layer data rate of 54 Mbps in the 2.4GHz band through the use of OFDM. While 802.11a and 802.11b are not interoperable, 802.11g allows for backwards compatibility towards 802.11b systems. Details on specifications for 802.11 a/b/g PHY and MAC specifications can be found in the respective standard documents, available online at [82].

While the 802.11 family of specifications are mainly geared towards infrastructure systems, it is possible to operate them in an ad hoc mode. In this case, stations (nodes) form a so-called Independent Basic Service Set (IBSS), in which any station within transmission range of another can initiate communication. Both single-hop and multihop communication can be realized with 802.11, making it possible to form small-scale MANETs.

There are, however, some problems associated with using 802.11 systems in ad hoc mode. In infrastructure mode, the Distributed Coordination Function (DCF) is used for providing CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), which effectively solves the medium access problem for a two-node ad hoc network. For larger non-trivial networks, the *hidden node* problem arises, as depicted in figure 2.4.2. Here, node B is within transmission range of both nodes A and C but the two latter nodes cannot hear each others' transmissions. Simultaneous transmission will lead to collisions and subsequent backoffs by both terminals, resulting in reduced throughput for all nodes. This is to a great extent avoided by a four-way handshake operation known as Request To Send/Clear To Send (RTS/CTS). When a node wishes to transmit a frame of data it sends an

RTS (containing the length of the requested channel occupancy) to the recipient node, which in turn replies with a CTS if transmission can be granted. The frame is then transmitted and ACK'ed by the receiver. The signaling of the transmission length within the RTS/CTS messages allows "hidden" nodes to wait for the corresponding time before attempting to request communication. However, the RTS/CTS strategy is not ideal, as it will sometimes inhibit communication that could have been successful and will not always prevent collisions, as discussed in e.g. [83] and [84].

Capacity of ad hoc networks has been reported to be an issue. In [85], the authors investigate 802.11 based MANETs under different traffic patterns, node densities and node patters/topologies. It is found that locality of traffic is a crucial factor for allowing scalability of MANETs to larger networks. This is mainly due to the observed negative impact on capacity when longer node chains (many hops used for relaying packets) are used.

Part A

Rate-distortion optimized packet loss protection for multichannel video communication

The main ideas and results of this part of the thesis have been published as follows:

Stian Johansen, Anna Kim, Bjørnar Libæk and Andrew Perkis — "On the Tradeoff between Complexity and Performance of error protection schemes for embedded codes over parallel packet erasure channels". In *Proceedings of the Norwegian Signal Processing Symposium (NORSIG-05)*, Stavanger, Norway, September 2005.

Stian Johansen and Andrew Perkis — "Unequal error protection for embedded codes over parallel packet erasure channels". In *Proceedings of IEEE Workshop on Multimedia Signal Processing (MMSP-05)*, Shanghai, China, October 2005.

Chapter 3

Rate-distortion optimized packet loss protection for multichannel video communication

This part of the thesis deals with extensions to known algorithms for error/loss protection of compressed video. Whereas earlier work on this topic has mainly considered the single-channel case, *the novelty of this work is the investigation of the case where multiple parallel channels with possibly different loss characteristics are used*. The notion of a 'channel' in this case is not limited to the case of different physical channels, but also logically different channels which may share a common transmission medium. Such scenarios are becoming more relevant with the gradual deployment of priority-enabled networks (e.g. DiffServ, differentiated services), as well as the usage of multiple path streaming (e.g. in mobile ad hoc networks). Algorithms for allocation of channel codes for rate-distortion optimized multichannel packet loss protection are presented. The algorithms are compared in terms of PSNR (peak signal-to-noise ratio) performance over a selected set of configurations and channel conditions. An analysis of the computational complexity of the different approaches is also given, leading to a basis useful for selecting which algorithms to use in different applications and scenarios.

3.1 Introduction

A potential obstacle for delivery of compressed audiovisual data over packet-based networks using unreliable protocols such as UDP are the often unavoidable packet losses (erasures). Unequal packet loss protection, as briefly introduced in section 2.2.2, is a family of techniques where the fundamental idea is that the degree of loss protection applied to a section of data should be in accordance with the relative importance of this particular data section. That is, more important data should be given a stronger protection than less important data. This of course also holds in the multichannel case, but the problem of optimally allocating channel codes in order to achieve the best possible rate-distortion performance becomes more complex. In the following, a brief overview of related work is given before the problem statement and the proposed algorithms are presented. Complexity is treated in some detail before results are given together with an evaluation of the tradeoff between complexity and performance for the discussed algorithms.

3.1.1 Related work

As mentioned in section 2.2.2, packet loss protections schemes inspired by the 'Priority Encoding Transmission' (PET) system originally proposed by Albanese et. al. [10] have been extensively studied in recent years. The PET system creates a channel coded packet stream from a partitioned (segmented) version of the encoded source data. A certain segment can then be decoded from a subset of packets given that the fraction of packets received is greater than or equal to a user-specified *priority value* for the segment. In this way, a high priority value implies that less packets are needed for reconstruction than for a lower priority value. The first algorithm for optimally determining these priority values is due to Danskin and Davis [86].

The concept of PET was later generalized and applied to embedded (i.e. progressive) data in [87] and [47]. The resulting techniques are commonly known as UXP (Unequal error/loss Protection), MD-UXP (as described in section 2.2.2 or even MD-FEC (Multiple Description coding based on Forward Error Correction); throughout the remainder of this thesis they will be referred to as UXP. The original single-channel formulation is similar to the upper third of figure 3.1. The different *codewords* are the rows of the figure, and consist of a certain number of channel code blocks (f_i, white blocks) and a number of source data blocks ($N - f_i$, grey blocks). Note that each 'block' of data can in general be of arbitrary size B. Packets are found as the columns of the figure, and have lengths given by LB where L is the number of codewords per channel. The data from the embedded source coder is distributed from codeword 1 to codeword L. That is, codeword 1 contains the $(N - f_1)$ most important data blocks, codeword 2 contains the following $(N - f_2)$ data blocks and so on, as indicated by the dashed arrowed line in the figure. Using an MDS (maximum distance separable,

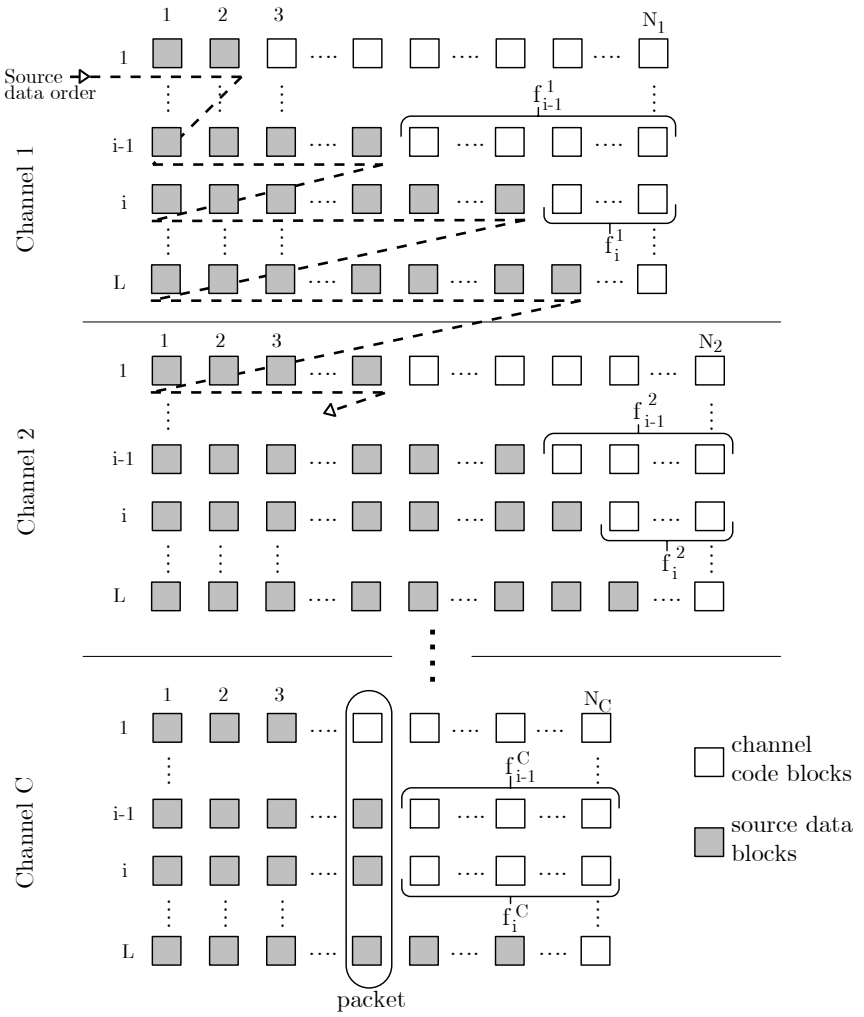


Figure 3.1: Conceptual distribution of information and channel code blocks.

see section 2.2.2) channel code (e.g. Reed-Solomon), codeword i can be correctly decoded given that the number of packets lost is no greater than f_i [10].

However, the computational complexity of optimally allocating channel codes in the above described scheme may be prohibitive in practice. Referring to figure 3.1, the number A of possible different allocations for the single-channel case

(upper third of the figure) is given by

$$A = L \sum_{n=1}^{N_1-1} (2n-1). \tag{3.1}$$

For each of these A allowable allocations the resulting expected distortion D (explained later) must be computed, in order to find the allocation yielding lowest distortion. For realistic communication rates, the computational complexity of this quickly becomes unwieldy.

To this end, several faster algorithms have been proposed in literature. In [47], a framework based on the method of Lagrange multipliers is used to find the optimal solution. A faster heuristic algorithm is then developed to approximate the optimal solution. Other optimum-attaining solutions are found in [88] and [89], of which the latter also proposes a faster algorithm that approximates the optimal allocation. The fastest proposed algorithm is due to Stankovic et.al. [90]. This algorithm achieves near-optimal results with a complexity of $O(NL)$ in the single channel case. This algorithm is used as the starting point for the work presented in this chapter, however modified and generalized to support the multichannel case. Thus, the notation used here is similar to that of [90].

A somewhat related approach to what is considered in this work is the use of unequal packet loss protection techniques for multicasting layered video to heterogeneous receivers. In [91] and [92], the authors consider the case where a high-bandwidth and a low-bandwidth client with different resulting packet losses share a bottleneck link. The main aim of these approaches is to minimize the quality loss that stems from the differing loss characteristics for either client. In this work this problem is not explicitly considered; it is rather assumed that a single client receives different parts (i.e. layers) of video over physically or logically different channels.

3.1.2 Applications

DiffServ enabled networks

The DiffServ [43] architecture was proposed in order to provide a more reliable and/or delay aware service to certain services when network utilization approaches capacity. In contrast to the per-flow fine-grained QoS (Quality of Service) architecture IntServ (integrated services) [93], DiffServ is a coarse-grained class based QoS provisioning architecture. Packet marking (using the so called DS codepoint) at sources or at edge routers is used to select per-hop behaviors (PHB) for aggregates of traffic. A PHB that is relevant for video traffic is the Assured Forwarding (AF) group. The AF PHB defines four independently forwarded traffic classes where three packet drop precedences can be assigned within each class. Roughly speaking, a congested DiffServ node will attempt to alleviate congestion by first dropping packets marked with a higher drop

precedence value. If congestion persists over a timescale larger than the end-to-end RTT for a flow, feedback can be used to inform the source of resulting loss/delay characteristics for the different parts of the flow. These can in turn be seen as different "channels" by the source, as different priority markings imply different losses and delays. The approach presented here can thus be applied to minimize distortion resulting from dropped packets, given limitations to transmission rate as indicated by congestion control schemes.

Multipath networks

In order to achieve higher throughput and/or a more outage-resilient service, multi-path video streaming has been proposed. This may be especially useful in MANETs (as discussed in greater detail in chapter 6), since throughput along a single path may be too low to achieve acceptable video quality. Different paths can exhibit significantly different loss/delay characteristics, calling for optimization of loss protection to the different parts (layers/channels) of the video.

Finally, it is mentioned that the approach presented in this chapter can be used (and is showcased as an example) for realizing the framework presented in chapters 4 and 5 of this thesis.

3.1.3 Chapter outline

The chapter is structured as follows. Section 3.2 formalizes the problem and goes on to describe the developed and investigated algorithms for solving the optimization problem. Section 3.4 describes source and channel models that are used in the simulations, and section 3.3 investigates the computational complexity of the algorithms. Simulation results are given in section 3.5, followed by a discussion in section 3.6. Finally, concluding remarks are given in section 3.7.

3.2 Optimization formulation

The problem of transmitting an embedded source code over a set of C parallel packet erasure channels is considered. The packet length L (columns in figure 3.1) is fixed for all channels, but the number of packets N_j (rows in figure 3.1) to be sent over each channel may vary. The latter corresponds to the codeword length N_j for channel j. These codewords constitute f_i^j channel code symbols and $(N_j - f_i^j)$ information (source) symbols. Thus, each individual codeword may have a different *code rate*, given by $(N_j, N_j - f_i^j)$. Here and in the following, the indices $i \in [1, L]$ and $j \in [1, C]$ indicate codeword number and channel number,

respectively. Within each channel, the *embedding property* is enforced:

$$f_1^j \geq f_2^j \geq f_3^j \geq \ldots \geq f_L^j \quad \forall\, j \in [1, C]. \tag{3.2}$$

That is, the code rate should be *nondecreasing* with increasing i. This property does not hold across channel boundaries (from codeword L in channel j to codeword 1 in channel $j+1$), since differing channel statistics may result in a stronger channel code being appropriate in channel $j+1$. The expected distortion D is given by

$$E[D] = \sum_{j=1}^{C} \sum_{i=0}^{L} \mathbf{P}(i,j) D\left(\mathbf{R}(i,j)\right). \tag{3.3}$$

Thus, the optimization problem is to find the channel code allocation that minimizes (3.3). The probabilities for correctly decoding codeword i in channel j are given in the probability matrix $\mathbf{P}$ of size $(L+1) \times C$. The expressions for finding the elements of this matrix are given in equation (3.4).

$$\mathbf{P}(i,j) = \begin{cases} \sum_{n=f_i^j+1}^{N_j} p_{N_j}(n) & i = 0,\ j = 1 \\ \sum_{n=f_{i+1}^j+1}^{f_i^j} p_{N_j}(n) & i \in [1, L-1], j = 1 \\ \left[\sum_{n=0}^{f_i^j} p_{N_j}(n)\right] \sum_{n=f_1^{j+1}}^{N_{j+1}} p_{N_{j+1}}(n) & i = L,\ j = 1 \\ 0 & i = 0, j \geq 1 \\ \Omega \sum_{f_{i+1}^j+1}^{f_i^j} p_{N_j} & i \in [1, L-1], j \geq 1 \\ \Omega \left[\sum_{n=0}^{f_i^j} p_{N_j}(n)\right] \sum_{n=f_1^{j+1}}^{N_{j+1}} p_{N_{j+1}}(n) & i = L,\ j \in [2, C-1] \\ \Omega \sum_{n=0}^{f_i^j} p_{N_j}(n) & i = L,\ j = C \end{cases} \tag{3.4}$$

where

$$\Omega = \prod_{k=1}^{j-1} \sum_{n=0}^{f_L^k} p_{N_k(n)}. \tag{3.5}$$

The elements $\mathbf{R}(i,j)$ of the rate matrix $\mathbf{R}$ give the (source) rate up to and including codeword i of channel j. Its elements are found as

$$\mathbf{R}(i,j) = \sum_{k=1}^{j-1} \left(L \cdot N_k - \sum_{l=1}^{L} f_l^k \right) + i \cdot N_j - \sum_{k=1}^{i} f_i^k. \tag{3.6}$$

3.2.1 Rate-optimal allocation

The *rate-optimal* (RO) allocation of channel codes is the allocation that maximizes the expected number of received source bits rather than minimizes the expected distortion. This solution is denoted RO-EXP, where EXP indicates

$\{f_1^1+1,\ f_2^1,\ f_3^1\cdots\}$
$\{f_1^1+1,\ f_2^1+1,\ f_3^1\cdots\}$
$\vdots$
$\{f_1^1+1,\ f_2^1+1,\cdots f_L^1+1,\ f_1^2+1,\ f_2^2\cdots\}$
$\vdots$
$\{f_1^1+1,\ f_2^1+1,\cdots f_{L-1}^C+1,\ f_L^C+1\}$

Table 3.1: All possible "neighborhoods" from an initial allocation given by $\{f_1^1, f_2^1, \cdots, f_L^1, f_1^2, \cdots, f_L^C\}$.

that this is a form of equal packet loss protection within each channel. As is conjectured in [94], the distortion-optimal *unequal* loss protection can never have a lower protection (i.e. weaker protection of any codeword) than the RO-EXP solution. Furthermore, the RO-EXP solution can be found in linear time. This fact forms the basis of the fast local search algorithm of [90], as the algorithm uses the RO-EXP solution as a starting point for finding the UXP solution. In the multichannel case considered in this work, the RO-EXP allocation is found as

$$f_i^j = \arg \max_{k=1,\cdots,N_j-1} \left\{ (N_j - k) \sum_{n=0}^{k} p_{N_j}(k) \right\} \ \forall j \in [1, C]. \tag{3.7}$$

3.2.2 Neighborhood definitions and optimization algorithms

In [90], the *neighborhood* $\mathcal{N}$ of a specific allocation $\mathbf{F}$ describes the set of all possible channel code allocations in which the error protection of codewords is increased by one 'block' (see figure 3.1) per codeword. The practical relevance is that the set of possible allocations in $\mathcal{N}$ are those that are investigated by the iterative optimization algorithms presented in the following.

Multichannel neighborhood definition

A direct extension of the single-channel neighborhood definition to the multichannel case is shown in table 3.1. Starting from a specific allocation $\mathbf{F} = \{f_1^1, \cdots, f_L^C\}$, the set of neighborhoods $\mathcal{N}$ is found by increasing the error protection of the first codeword of the first channel, and then successively to higher numbered codewords and successively higher numbered channels. Note that the embedding property ($f_i^j \geq f_{i+1}^j \forall\ i, j$) is automatically satisfied using this neighborhood definition. With this extension of the neighborhood definition, the optimization algorithm of [90] can be used more or less directly. The resulting optimization algorithm is outlined in algorithm 1. In the following, this approach will be referred to as MCN (MultiChannel Neighborhood).

Algorithm 1 Using multichannel neighborhood definition (MCN)

```
1: Compute rate-optimal solution according to equation 3.7 and set this
   allocation as F_best.
2: set MSE_min = MSE of rate-optimal solution via equation 3.3
3: while MSE_min lowered during last iteration do
4:    Set allocation F to F_best
5:    for c = 1 to C do
6:       for l = 1 to L do
7:          Set f_l^c = f_l^c+1
8:          Find resulting expected MSE_current via equation 3.3
9:          if MSE_current < MSE_min then
10:            Store allocation as F_best
11:            Set MSE_min = MSE_current
12:         else
13:            Break out of both 'for' loops
14:         end if
15:      end for
16:   end for
17: end while
```

However, as will be evident from the simulation results, the performance achieved by the MCN approach decreases noticeably when the number of channels increases. The reason for this is that the neighborhood definition is inherently oblivious to the characteristics of the different channels (the p_{N_j}s in equation 3.4). Specifically, the neighborhood definition does not allow strengthening the error protection for a codeword in channel j without first strengthening the error protection for all codewords in channel $j-1$.

Sequential optimization

Bearing this in mind, a plausible alternative is to do the optimization sequentially, starting from the lowest channel. That is, first optimizing the lowest channel ($j = 0$) while keeping the others unchanged, then optimizing the second channel while keeping the others unchanged, and so on. Effectively, the neighborhood definition of [90] is then used directly. Unfortunately, the performance of this scheme (as results will show) is not an improvement on the MCN approach. The main flaw of this approach is that the optimization of error protection for a specific channel will be done using suboptimal allocations for the other channels. This leads to a bias towards too strong error protection for the channel(s) that are optimized first, as the optimization procedure will attempt to compensate for the (non-optimized) weaker error protection for subsequent channels. The outline for the sequential optimization approach is given in algorithm 2, and will

be referred to as SEQ (i.e. 'sequential') in the simulation results.

Algorithm 2 Sequential optimization using single-channel neighborhood definition ('SEQ')

Compute rate-optimal solution according to equation 3.7; store this as $\mathbf{F}_{best}$.
Set MSE_{min} = MSE of rate-optimal solution via equation 3.3
for $c = 1$ to C **do**
  while MSE_{min} lowered during last iteration **do**
    Set allocation $\mathbf{F} = \mathbf{F}_{best}$
    for $l = 1$ to L **do**
      Set $f_l^c = f_l^c + 1$
      Find resulting expected $\text{MSE}_{current}$ via equation 3.3
      if $\text{MSE}_{current} < \text{MSE}_{min}$ **then**
        Store allocation as $\mathbf{F}_{best}$
        Set $\text{MSE}_{min} = \text{MSE}_{current}$
      else
        Break out of innermost 'for' loop
      end if
    end for
  end while
end for

Structured sub-search

In order to achieve a better multichannel performance than the MCN and SEQ algorithms, a more rigorous search approach is needed. However, as the computational complexity (detailed discussion in section 3.3) quickly becomes unwieldy when a more exhaustive search approach is employed, it would be beneficial to still use a variant of the neighborhood approach from [90]. To this end, a structured approach that can be implemented as a recursion is proposed. Instead of using the multichannel neighborhood definition of the MCN algorithm outlined above, the original single channel neighborhood definition is used. During the optimization procedure, for each element of $\mathcal{N}_j$ for channel j, the search algorithm is run for elements of $\mathcal{N}_{j+1}$ in channel $j+1$ and so on recursively until $j = C$ is reached. Thus, the full range of possible neighborhoods are investigated. A visualization of the relationships between the different neighborhoods considered during optimization are shown in figure 3.2 for the simple case of $L = 2$ and $C = 3$. An outline of the optimization algorithm, implemented as a recursion, is given in algorithm 3. Note that, in order to reduce the number of required computations, the algorithm (like algorithms 1 and 2) starts from the rate-optimal solution given by equation 3.7. This algorithm will be referred to as SSS.

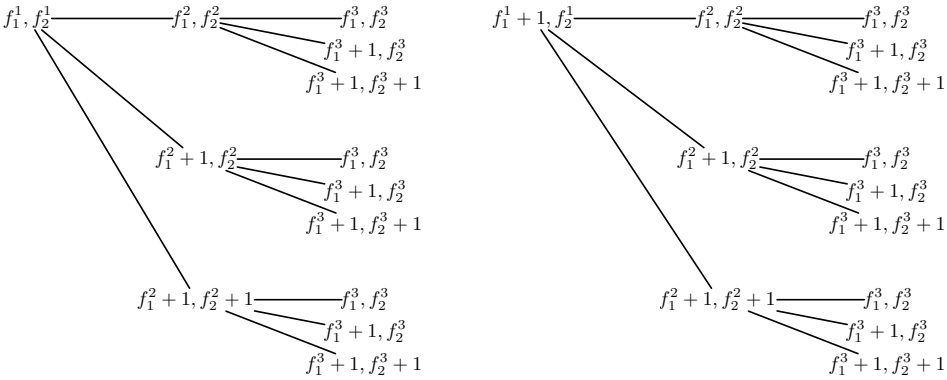


Figure 3.2: Relationships between the different neighborhoods as used by the structured sub-search (SSS) algorithm. For readability, the alternatives for the (f_1^1+1, f_2^1+1) allocation in channel 1 are not shown.

Algorithm 3 Recursive optimization using structured sub-search ('SSS')

Require: The initial allocation matrix $\mathbf{F} = \{f_i^j\} \forall i \in [1, L], j \in [1, C]$.
Require: The channel number c which is to be considered
while MSE_{min} improved during last iteration **do**
  for $l = 1$ to L **do**
    Set $f_l^c = f_l^c + 1$
    if $c < C$ (the channel considered is not the last one) **then**
      Call 'SSS' algorithm with $c = c + 1$ and the modified allocation $\mathbf{F}$
      Store the returned $\mathbf{F}$ from recursive call of 'SSS'
    end if
    Find resulting expected $\text{MSE}_{current}$ using eq. 3.3
    if $\text{MSE}_{current} < \text{MSE}_{min}$ **then**
      Set $\mathbf{F}_{best} = \mathbf{F}$
      Set $\text{MSE}_{min} = \text{MSE}_{min}$
    else
      Break out of 'for' loop
    end if
  end for
end while
return The best found allocation, $\mathbf{F}_{best}$

RD-optimized equal error protection

A conceptually and computationally simpler approach is to employ equal loss protection for all codewords contained in a channel. As mentioned, the rate-optimal (RO-EXP) solution (which is used as a starting point in the presented optimization algorithms) is an example of an equal error protection scheme. However, the rate-distortion performance of the 'RO' approach is significantly worse than all of the other presented optimization algorithms. This is to be expected, as the calculation of the RO-EXP solution does not consider the rate-distortion characteristics of the source in any way.

By finding the best *equal* loss protection allocation in a rate-distortion sense, performance close to the *unequal* error protection case can be found with significantly lower computational complexity. The algorithm outline given in algorithm 4 searches for the best such solution where the code rate is maintained equal for all codewords within a channel. Note that the number of packets N_j within each channel $j \in [1, C]$ may vary. This algorithm will be referred to as 'RDE' in the remainder of this chapter.

Algorithm 4 Rate-distortion optimized equal error protection ('RDE')

Compute rate-optimal solution according to equation 3.7; store this as $\mathbf{F}_{best}$.
Set MSE_{min} = MSE of rate-optimal solution via equation 3.3
while $n < N$ **do**
  for $c = 1$ to C **do**
    if $n <$ {number of packets in channel c} **then**
      Set $f_l^c = n \; \forall l \in [1, L]$
      Find resulting $\text{MSE}_{current}$ via equation 3.3
      if $\text{MSE}_{current} < \text{MSE}_{min}$ **then**
        Set $\mathbf{F}_{best} = \mathbf{F}$
        Set $\text{MSE}_{min} = \text{MSE}_{current}$
      else
        Set $\mathbf{F} = \mathbf{F}_{best}$
      end if
    end if
  end for
  Set $n = n + 1$
end while

3.3 Complexity

In this section, the computational complexity of the algorithms presented earlier is discussed. For quantifying the complexity, *polynomial time* analysis

(specifically, the O-notation) is used. The reader is referred to [95] for details on polynomial time analysis.

3.3.1 Single-channel UXP

As reported in [90], the single-channel algorithm has a worst-case running time of $O(NL)$, where N and L are the numbers of packets and codewords, respectively. The calculation of the rate-optimal (RO-EXP) solution is $O(N)$. The number of refinement iterations is upper bounded to $N-1$. At each iteration, the cost function needs to be calculated and compared to the other results at most L times (the maximum number of elements in the neighborhood). The calculation of the cost function has a computational cost bounded as $O(N)$. For all candidates in the neighborhood, a certain (single) codeword has been assigned an additional channel code block (added protection) compared to the 'previous' considered candidate. The decoding probabilities that need recalculation are those corresponding to the changed codeword and the one immediately following (this can be seen from equation 3.4). Since only two of the decoding probabilities change for each iteration, the computation of the cost function does not contribute significantly to the running time of the algorithm. Thus, the resulting worst-case running time is $O(NL)$. As will be evident, the complexity of computing the cost function is not equally insignificant in the multichannel case.

3.3.2 Multichannel UXP variants

When going from a single channel to multiple channels, the number of packets N in the channel is no longer fixed, as the N_j $j \in [1, C]$ are allowed to vary. However, in order to have a more unified notation, the subscript j in N_j is dropped for complexity considerations, despite the slight abuse of notation that this implies. Notice also that, if the packet size is maintained constant, the relation $\sum_{j=1}^{C} N_j = N$ holds if the transmission rates in the multichannel and single-channel cases are the same. With the assumption that the number of packets per channel is equal, the total number of packets CN_{multi} in the multichannel case is equal to the N_{single} packets of the single-channel case. This should be kept in mind when assessing the complexity considerations in the following.

The rate-optimal RO-EXP allocation is used as the starting point for all the multichannel optimization algorithms presented here. Since the computation of the rate-optimal solution is done independently for each channel (see equation 3.7), this algorithm has a running time characterized by $O(CN)$.

Another common complexity factor of the multichannel algorithms concerns the calculation of the $\mathbf{P}$ matrix (see equation 3.4). In the single-channel case, only two elements of the $\mathbf{P}$ matrix need to be recomputed at each step when evaluating each member of the neighborhood. This led to the conclusion that the complexity associated with recomputing the cost function at each specific

element of the neighborhood set $\mathcal{N}$ was negligible. This does, however, not hold in the multichannel case. In this case it is necessary to update the decoding probabilities of all codewords in subsequent channels if the very last codeword (number L) in a channel has changed in the current iteration. This is evident from the Ω-factor in the expressions for the decoding probabilities for higher-numbered channels ($j \geq 1$) in equation 3.4. Furthermore, it is necessary to update the decoding probability of the last codeword in the previous channel if the first codeword of a higher-numbered channel has been updated. Although these somewhat complicating cases happen relatively seldom at run-time, they do increase complexity in the O-sense. Specifically, up to $L(C-1)+1$ elements of the $\mathbf{P}$ and $\mathbf{R}$ matrices may need to be recomputed, along with the resulting updated calculation of equation 3.3.

The 'SSS' algorithm

In the case of the 'SSS' algorithm, the number of elements in the neighborhood set $\mathcal{N}$ is $(L+1)^C$. This can be seen from from figure 3.2, where the structure forms a $(L+1)$-ary tree. Note, however, that not all of these neighborhood elements will end up being investigated, since the algorithm breaks out of the search process if the resulting MSE does not improve. Nevertheless, in terms of O-notation the algorithm is characterized by $O(NCL^{C+1})$ when taking into account the worst-case cost of recalculating $\mathbf{P}$ and $\mathbf{B}$.

The 'MCN' algorithm

In this case the neighborhood set $\mathcal{N}$ is smaller than in the 'SSS' case. More precisely, the number of elements in $\mathcal{N}$ is CL. With the number of packets per channel being N, the running time is bounded as $O(NL^2C^2)$.

The 'SEQ' algorithm

The running time of this variant is bounded in the same way as the 'MCN' algorithm. This is due to the fact that the worst-case running time for the optimization of each channel is $O(NL^2C)$ but it needs to be done C times in succession.

The 'RDE' algorithm

The equal error protection approach of the 'RDE' algorithm will at most do NC computations of $\mathbf{P}$, $\mathbf{B}$ and the cost function of equation 3.3. Thus, the running time of this algorithm is bounded as $O(NLC^2)$. Compared to the above mentioned algorithms, this is a notable complexity reduction for realistic values of N, L and C.

Algorithm	Complexity
SSS	$O(NCL^{C+1})$
MCN	$O(NL^2C^2)$
SEQ	$O(NL^2C^2)$
RDE	$O(NLC^2)$

Table 3.2: Summary of polynomial-time complexity properties of the four investigated algorithms.

For reference, table 3.2 summarizes the polynomial-time complexities of the different algorithms.

3.4 Source and channel models

3.4.1 Source model

The distortion-rate function $D(R)$ needs to be modeled in order to perform the optimization outlined in section 3.2. In this work the the motion compensated embedded zero-block coding (MC-EZBC) video coder [38] (as described in chapter 2.1.3), is used as source model. The software was compiled from source using the latest version (July 2005) incorporating developments from RPI[1] and RWTH[2], as available from [96]. It is however emphasized that any embedded source coder can be used. The applicability of the proposed scheme for more coarse-grained scalable coders (such as e.g. the SVC extension of H.264/MPEG-4 AVC) is commented on in section 3.6.

Details on the parameter settings for the MC-EZBC encoder as well as video sequence information is given in table 3.3.

To approximate the distortion-rate function function with a parametric model, the Foreman video sequence in CIF (Common Intermediate Format) resolution is encoded at a high rate. Sub-bitstreams at lower rates are then extracted at rates ranging from 90 kbps to 2500 kbps, and subsequently decoded to find the resulting averaged MSE (Mean Squared Error). Note that only the luminance (Y) component is used for MSE modeling, averaged over all frames. It is noted that, in a practical implementation, more accurate modeling could be achieved by making the parametric model adaptive. That is, modeling should be done on a per-GOP (group of pictures) basis. Using least-squares curve fitting, the parametric model given by $\hat{MSE}(r)$ in equation 3.8 is used. The calculated parameters approximating the measured values from the video coder are given in

[1] Rensselaer Polytechnic Institute, NY, USA
[2] Rheinisch-Westfälische Technische Hochschule, Aachen, Germany

Video test sequence	*"Foreman"*
Number of frames	300
Frames per second	30
Resolution	*CIF* (352x288 *pixels*)
De-noising	*Disabled*
Temporal decomposition levels	4 (*GOP size* 16)
Macro block size	64x64 *pixels*
Motion vector accuracy	*quarter pixel*

Table 3.3: Video sequence details and the most important encoding parameters used for modeling of the distortion-rate performance.

k_1	2243.6	λ_1	92.3
k_2	10.8	λ_2	1.8
k_3	7.4	λ_3	15.0

Table 3.4: Parameters for the $\hat{MSE}(r)$ parametric model given in equation 3.8.

table 3.4. It is mentioned that the choice of parametric model is along the lines of [97].

$$\hat{MSE}(r) = k_1 e^{-\lambda_1 r} + k_2 e^{-\lambda_2 r} + k_3 e^{-\lambda_3 r}. \tag{3.8}$$

Converted to PSNR for visualization purposes, the parametric model and the measured data points for the video coder in question are plotted in figure 3.3. As is evident from the figure, the parametric model closely describes the rate-distortion performance of the video coder.

3.4.2 Channel models

For modeling the behavior of a lossy packet network, where packets may be lost because of congestion or physical losses (i.e. corruption over a wireless link), two models are used in this work. These are the Gilbert model [98] and the more commonly used binomial loss model.

Packet losses typically occur in bursts, calling for modeling of the channel using a burst-loss model. The simplified Gilbert model [98] [99] is such a burst-loss model. The model is based on the two-state Markov model shown in figure 3.4, where it is seen that the model can be in a good (G) and bad (B) state. The probabilities for a loss when in the G and B states are denoted P_G and P_B, respectively. In the simplified Gilbert model (used in this work), $P_G = 0$ and $P_B = 1$. That is, there is always a loss in the B, state but never a loss when in

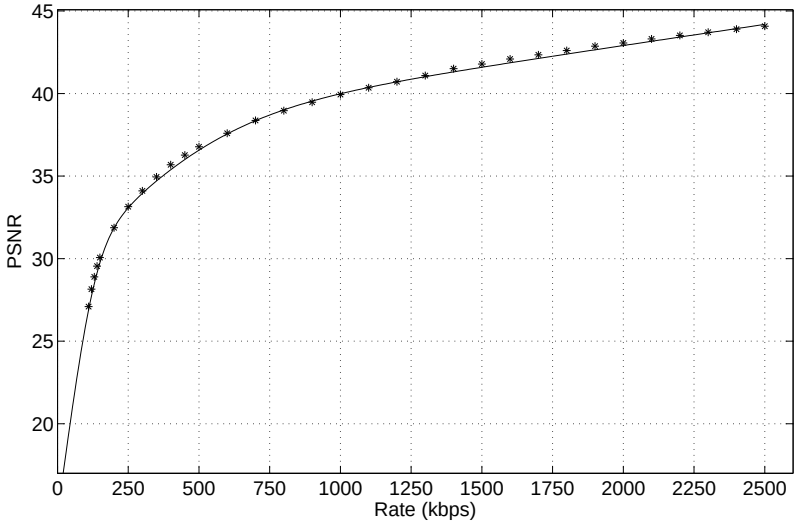


Figure 3.3: The found parametric model $\hat{MSE}(r)$ (solid line) and the data points used for least squares curve fitting. Note that the MSE values have been converted to PSNR for visualization purposes.

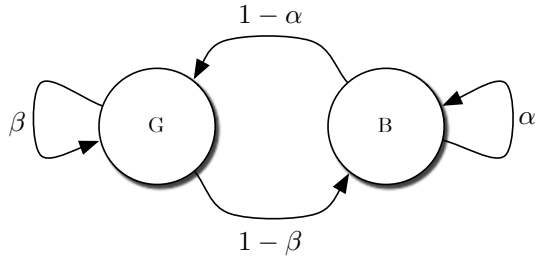


Figure 3.4: The Gilbert channel model.

the G state. The transition probabilities specified by α and β in turn determine the average loss probability p and the correlation between consecutive losses ρ as shown in equation set 3.9.

$$\begin{aligned} p &= \frac{1-\beta}{1-\alpha+1-\beta} \\ \rho &= \alpha+\beta-1 \end{aligned} \tag{3.9}$$

The distribution of packet losses $p_N(n)$ (probability of losing n out of N packets) is throughout its use in this thesis calculated using the iterative approach of [99].

The ρ parameter can be interpreted as a factor indicating the spread of the packet loss pdf.

The random packet loss model is also considered, in which case the distribution of packet losses is given by the binomial distribution, as given by equation 3.10. Again, N here denotes the number of packets.

$$p_N(n) = prob(n|N) = \left(\begin{array}{c} N \\ n \end{array} \right) p^N (1-p)^{N-n} = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n} \quad (3.10)$$

It is noted that the distribution resulting from the Gilbert model includes the binomial distribution as a special case in the limit where $\rho \to 0$. Conversely, when $\rho \to 1$, the Gilbert model yields a distribution where $p_N(n) \to 1/(N+1) \; \forall \; n \; \in [0, \cdots, N])$. Thus, as noted above, the ρ parameter can be seen as a measure of uncertainty in that increasing ρ yields a widening of $p_N(n)$.

3.5 Results

In this section, a selection of simulation results are presented. Emphasis is placed on the relative performance of the different algorithms. This is done in order to give a good basis for assessing the usefulness of the lower-complexity algorithms, as well as giving an indication of the performance of the schemes under different channel configurations and conditions. All the presented results share the following:

- Source model as described in section 3.4.
- All MSE results converted to PSNR for visualization.
- All simulations use the Gilbert distribution as introduced earlier, but with varying parameters. Details are stated in accompanying text and figure captions.
- In figures that give results as a function of packet loss fraction, the *weighted* average packet loss fraction is used. Since the packet loss fraction of the individual channels may vary, the average packet loss across channels is found through weighting, using the fraction of packets that are communicated over each individual channel. As an example, consider three channels communicating $[N_1, N_2, N_3]$ packets (N_{tot} in total) and having packet loss fractions $[p_1, p_2, p_3]$. The weighted average would then be found as $p_{wavg} = p_1(N_1/N_{tot}) + p_2(N_2/N_{tot}) + p_3(N_3/N_{tot})$. The number of packets and the individual packet loss fractions will be stated.
- Unless otherwise noted, the number of codewords per channel (L) is 10. This is done by adjusting the number of bytes per "block" in figure 3.1.

3.5.1 Performance at different transmission rates and under different channel conditions

Figures 3.5 - 3.9 give the expected decoded PSNR as a function of packet loss fraction for varying transmission rates and varying distributions of packet losses. The number of channels C is three in all simulations.

Figures 3.5 a)/b) and 3.6 a)/b) represent transmission rates of 300, 600, 900 and 1200 kbps, respectively, with the distribution parameter $\rho = 0.5$ for packet losses. The number of packets is equal in each channel (packet size 1000 bytes), and the packet loss fraction is increased by 0.01 for each channel (that is, $\frac{E[p_{N_{j+1}}(n)]}{N_{j+1}} = \frac{E[p_{N_j}(n)]}{N_j} + 0.01$).

Figure 3.7 a/b shares the above configuration, with the exception that both plots are for a transmission rate of 600 kbps, while having distribution parameters $\rho = 0.2$ and $\rho = 0.8$, respectively.

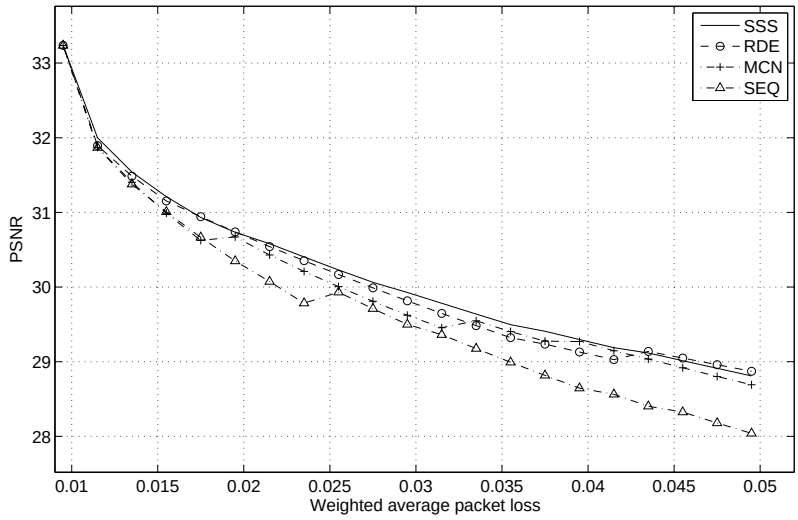
The results show that the SSS algorithm consistently yields best performance at all packet loss fractions, as is to be expected. The RDE and MCN algorithms perform within approx. 0.3dB compared to the SSS algorithm, while the SEQ algorithm tends to exhibit notably (up to 1dB) lower PSNR performance in some cases. There is a tendency towards a smaller difference between the algorithms (in terms of PSNR performance) as the transmission rate increases. The distribution parameter ρ is not seen to influence the above conclusions to a large extent, although having a noticeable influence on the attained PSNR values (as is evident from figures 3.7 a) and b). This is as expected, considering the influence of the ρ parameter as discussed in section 3.4.

The most important insight to be taken from these results is however that the performance difference between the algorithms in most cases is fairly minor, especially for lower packet losses. This observation is certainly in favor of choosing the lower complexity RDE algorithm for many applications.

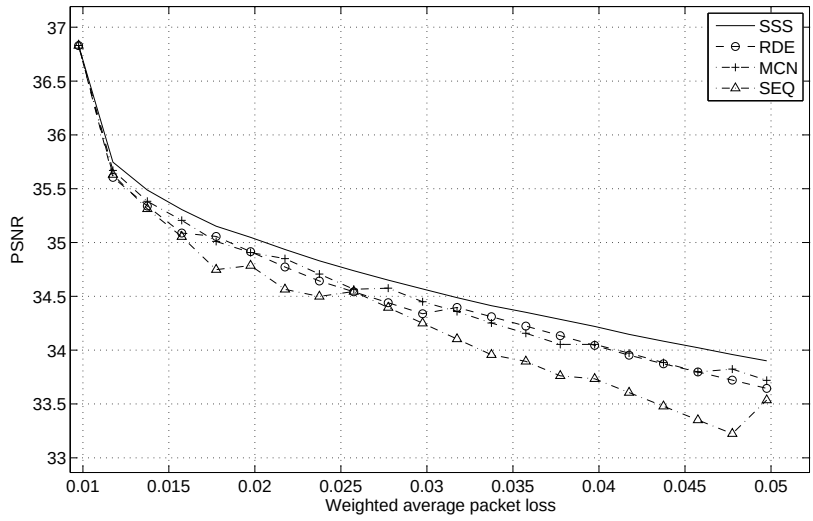
3.5.2 Unbalanced rates across channels

As mentioned, the results described above represent the case where the transmission rate in each channel is equal. Figure 3.8 a) and b) shows results where the rates of the different channels are unequal. Specifically, figure 3.8 a) shows the case of $[10, 20, 30]$ packets per GOP in the three channels, while figure 3.8 b) shows the opposite case; namely $[30, 20, 10]$ packets per GOP. This yields a total rate of 900 kbps when using 1000 byte packets. Packet loss fractions are as in section 3.5.1, using a Gilbert distribution parameter $\rho = 0.5$.

Results indicate that such an uneven distribution of transmission rate across channels increases the performance gap between the different algorithms. For the case of figure 3.8 a), it is clearly seen that the increase of transmission rate with increasing channel index is quite strongly in favor of the SSS algorithm, while the SEQ algorithm exhibits strongly reduced performance. It is also seen that

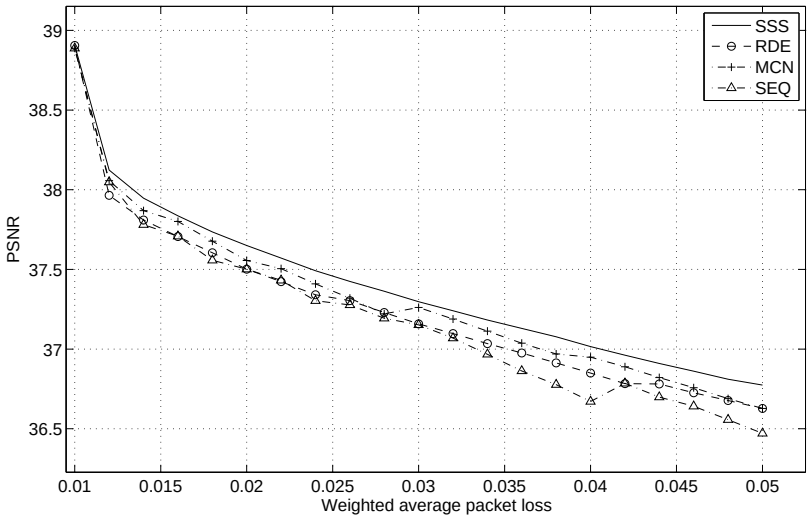


(a)

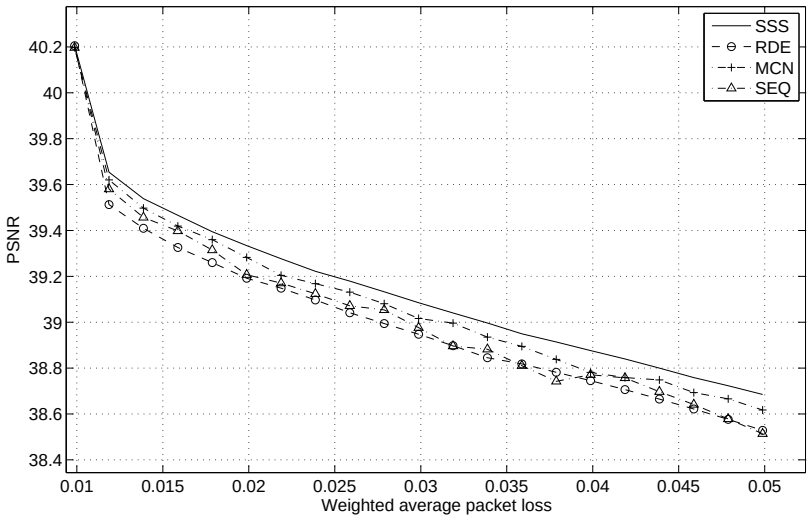


(b)

Figure 3.5: Performance of the different optimization algorithms for total transmission rates of 300 kbps (a) and 600 kbps (b). Three channels, packet size 1000 bytes, $\rho = 0.5$ and with packet loss fraction incremented by 0.01 per channel.

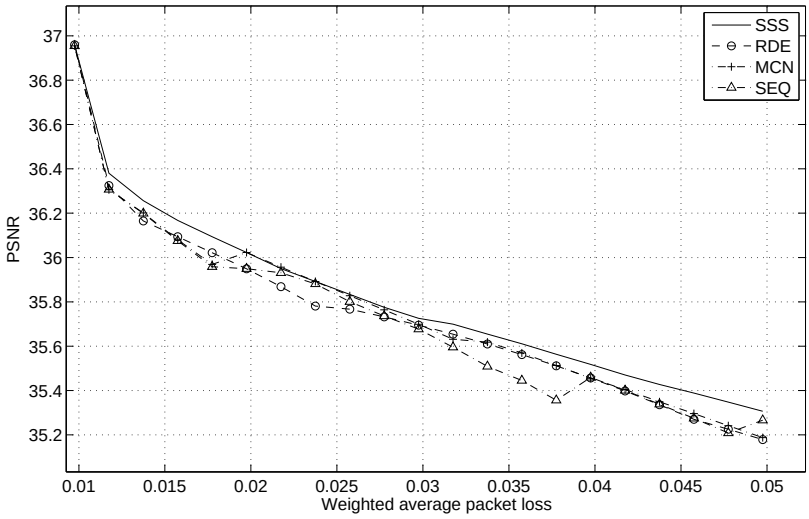


(a)

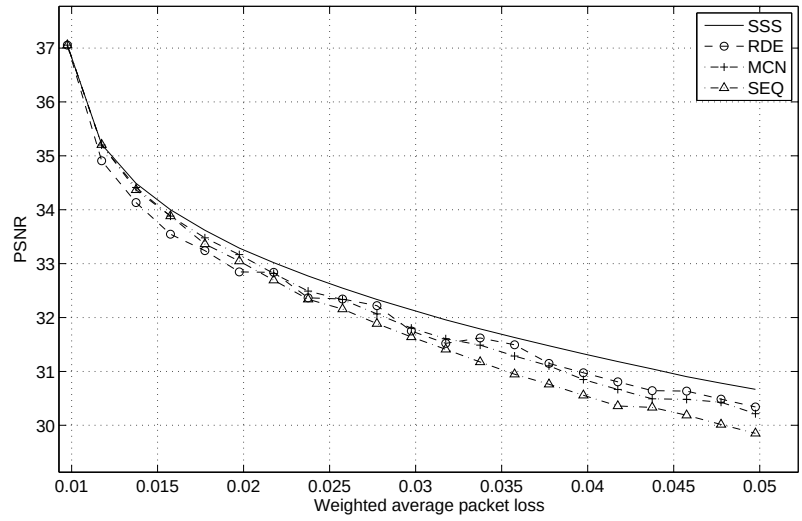


(b)

Figure 3.6: Performance of the different optimization algorithms for total transmission rates of 900 kbps (a) and 1200 kbps (b). Three channels, packet size 1000 bytes, $\rho = 0.5$ and with packet loss fraction incremented by 0.01 per channel.



(a)



(b)

Figure 3.7: Performance of the different optimization algorithms for $\rho = 0.2$ (a) and $\rho = 0.8$ (b). Three channels, total transmission rate 600 kbps, packet size 1000 bytes, and with packet loss fraction incremented by 0.01 per channel.

the RDE algorithm gives notably lower performance in this case compared to the SSS algorithm.

For the opposite case (figure 3.8 b)), algorithms SSS, MCN and SEQ perform almost identically, while the RDE algorithm is seen to yield a somewhat reduced performance (on the order of 0.2dB).

It thus seems clear that the choice of optimization algorithm should take into account the distribution of rates across channels for the application in question.

3.5.3 Packet size dependability

Figures 3.9 a) and b) show how the performance of the different proposed algorithms varies with packet size in the three-channel case. Total transmission rates are 900 and 400 kbps, respectively, communicating an equal number of packets per channel. The packet loss fractions of the three channels are $[0.2, 0.3, 0.4]$, all using $\rho = 0.5$ for the Gilbert channel model.

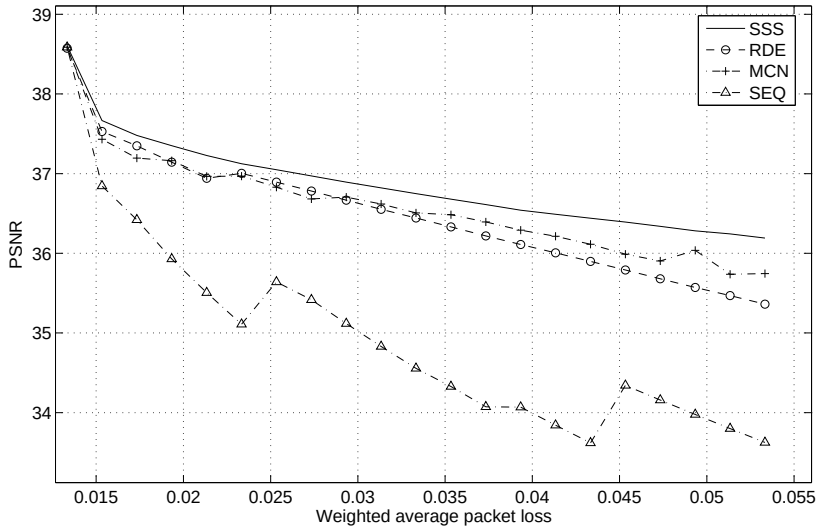
It can be seen that the performance gap between the different algorithms increases with increasing packet size. This is particularly true for the lower rate (400 kbps) case, where the SEQ algorithm deviates noticeably at higher packet sizes.

An important observation is that the performance generally decreases (for all algorithms) as the packet size increases. The main reason for this is that a larger packet size implies that fewer packets are sent (as the total transmission rate is kept constant). An implication of this is that the number of possible allocations of channel codes decreases, thus reducing the available degrees of freedom in optimization. Thus, this set of results indicate that choosing a smaller packet size is beneficial in terms of RD performance. The performance of the different considered optimization algorithms is furthermore almost identical at smaller packet sizes, an observation that calls for the use of the lowest complexity algorithm (RDE) in this case. If larger packet sizes are needed (or preferable because of overhead issues), a gain can be seen from using the SSS algorithm instead.

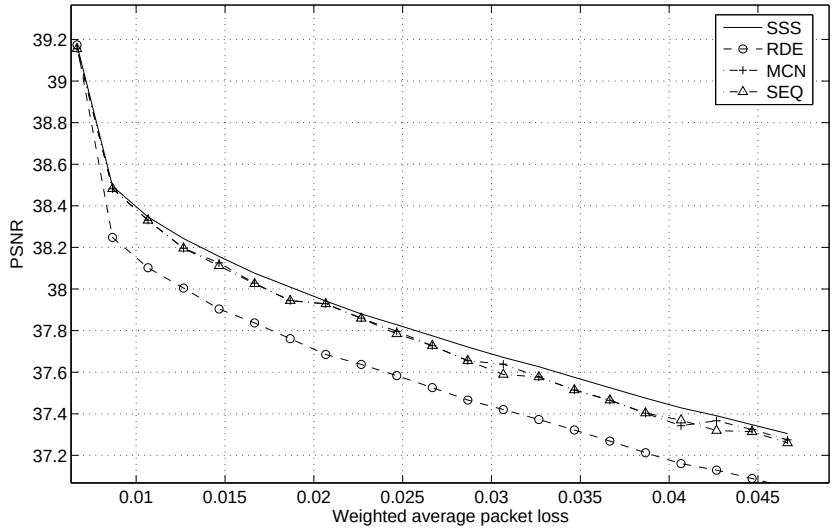
3.5.4 Influence of the number of channels

Figures 3.10 a) and b) highlight how the performance of the different algorithms develops as the number of channels increases. The results are for the cases of 675 and 1350 kbps, respectively, using a packet size of 1500 bytes and equal numbers of packets per channel. Packet loss of the first ("base") channel was 0.01, with an increase of packet losses in each channel increasing by 0.01 per additional channel (i.e. [0.01], [0.01, 0.02], [0.01, 0.02, 0.03] and so on).

It is seen from the results that the performance gain seen from using the SSS algorithm increases as the number of channels increases. An exception here is for the RDE algorithm, for which the performance gap is large for the single-channel

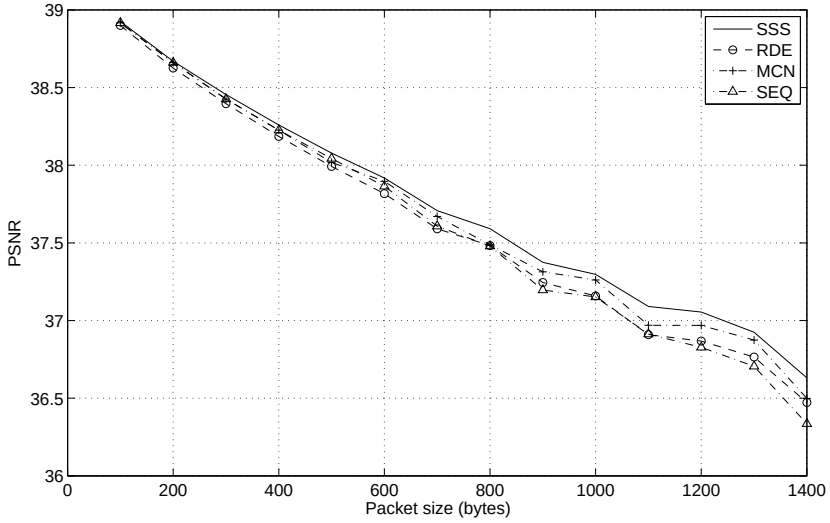


(a)

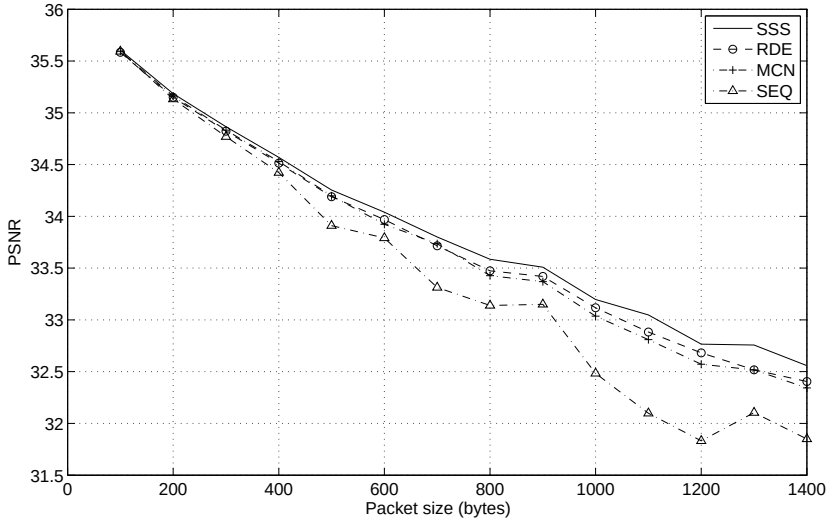


(b)

Figure 3.8: Performance of the different optimization algorithms when the number of packets per channel is uneven. Here, three channels communicate $[10, 20, 30]$ (a) and $[30, 20, 10]$ (b) packets per GOP. Total transmission rate is 900 kbps, packet size 1000 bytes, $\rho = 0.5$, packet loss fraction incremented by 0.01 per channel.



(a)



(b)

Figure 3.9: Performance of the different optimization algorithms various packet sizes. Total transmission rates are 900 kbps (a) and 400 kbps (b). $\rho = 0.5$, packet loss fractions $[0.02, 0.03, 0.04]$ in the three channels.

case but tends to approach the performance of SSS (and outperform the MCN and SEQ algorithms) as the number of channels increases. This effect is to a large extent explained by the fact that the number of packets per channel decreases as the number of channels increases (given a fixed total rate).

3.5.5 Number of codewords per channel

Thus far, the number of codewords per channel (given by the L parameter, see figure 3.1) has been kept constant at 10 for the presented results. Obviously, the L parameter along with the number of packets N_j per channel restricts the "granularity" of the optimization procedure. Specifically, a high value of L will allow a more precise allocation of error protection at the cost of higher computational complexity (see dependencies on L in section 3.3). Figure 3.11 compares the influence of the L parameter on the performance of the different optimization algorithms. The figure shows the case of a transmission rate of 675 kbps, having three channels carrying equally many 500-byte packets. The ρ parameter for the Gilbert model is here 0.5.

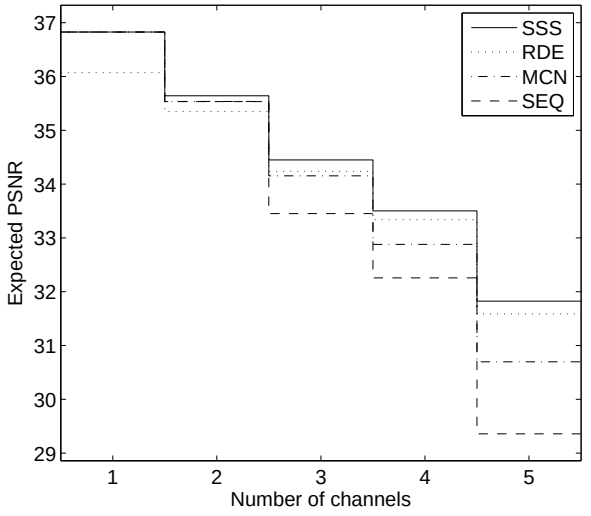
As is evident from the figure, the influence of the L parameter is limited. As is to be expected, the SSS and MCN algorithms perform identically with the RDE algorithm in the extreme case when $L = 1$. When L increases, the added flexibility for performing unequal protection gives room for increased performance, yielding better performance for the SSS algorithm in all cases and to a lesser extent also for the MCN algorithm. Naturally, the performance of the RDE algorithm does not depend on L.

An interesting result is that most of the attainable gain is achieved by a relatively low L (between 5 and 10), leading to the conclusion that the added complexity of increasing the L further is not justified. Perhaps surprisingly, the performance of the MCN algorithm actually decreases after experiencing a maximum at $L = 2$. This is related to the fact that increasing L increases the search space and thereby, roughly speaking, also increases the probability for arriving at a suboptimal solution. This effect is however not experienced for the SSS algorithm.

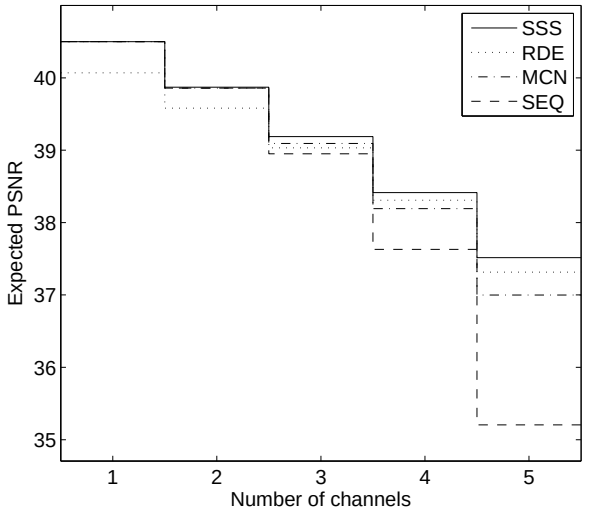
3.6 Summary and discussion

3.6.1 Algorithm selection

Simulation results indicate that the considered algorithms perform similarly in many scenarios. If complexity is an issue, this observation calls for the application of the RDE algorithm in most cases. There are however scenarios where this conclusion may be questioned. Specifically, the SSS algorithm has a noticeable advantage when the number of channels is high and/or the distribution of packets across channels is uneven. Furthermore, the gap between the SSS algorithm and



(a)



(b)

Figure 3.10: Performance of the different optimization algorithms for varying numbers of channels. Total number of packets per GOP is 30 (a) and 60 (b), resulting in total transmission rates of 675 and 1350 kbps, respectively. Packet size 1500 bytes and $\rho = 0.5$. Packet loss fraction of first channel is 0.01, and incremented by 0.01 per additional channel.

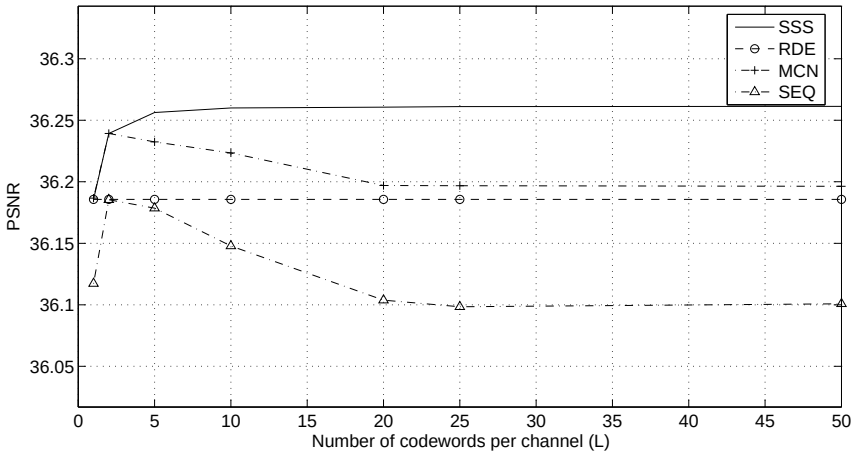


Figure 3.11: Performance of the different algorithms as a function of the number of codewords per channel L. Total transmission rate fixed at 675 kbps, packet size 500 bytes, three channels, $\rho = 0.5$ and packet loss fractions $[0.02, 0.03, 0.04]$.

the others increases as the packet loss fraction increases. It seems clear that the SEQ algorithm should not be used, as it exhibits inferior performance at a higher computational complexity than the RDE algorithm in all but one of the considered cases.

3.6.2 Source coder dependency

As described earlier, the source coder used in this work is the MC-EZBC coder. This class of coders lend themselves nicely to the unequal loss protection framework, since (conceptually) each additional decoded byte of data will improve the decoded PSNR. In any case, the smallest unit in the bitstream of an embedded code is orders of magnitude smaller than what is the case for most non-embedded coders.

However, as the coding efficiency of embedded video coders like MC-EZBC is somewhat inferior to their hybrid counterparts (as well as commercially unavailable), it would be beneficial if the unequal loss protection could be used for more coarse-grained hybrid scalable coders. This is more problematic in terms of modeling and flexibility, since the layers are (normally) fixed at encoding time, and the smallest decodable unit in hybrid layered coders is larger than in the embedded case. For layered video coders, the same level of error protection should be applied for all data contained in a specific layer. Since the number of layers is typically low in hybrid layered codes, this could potentially decrease the efficiency of unequal loss protection schemes. However, as results in section 3.5.5 indicate, this may not necessarily be the case. The number of loss protection

classes per channel is equal to L, and figure 3.11 indicates that the dependency on this parameter is low[3]. An intuitive approach in the multichannel case would be to have one video layer per channel. A complicating factor in the context of this work will arises when the total transmission rate per channel is fixed (as is the case considered here).

For considering unequal packet loss protection for hybrid video coders, the work in [100] deserves a mention. Here, the authors use the scalable extension (SVC) of H.264/AVC in combination with (single-channel) unequal loss protection. The work is based on the discontinued FGS (fine grain scalability) branch of SVC, but should be applicable also for the MGS (medium grain scalability) option of SVC.

3.7 Conclusions

This chapter has introduced extensions to the unequal packet loss protection framework for communicating over multiple parallel channels. Based on earlier proposed single-channel optimization algorithms, the work presents a set of algorithms that extend this to the multichannel case. The presented algorithms are compared in terms of both performance and complexity, leading to a basis for algorithm selection for specific applications and/or network configurations. The computationally most efficient algorithm (RDE) is seen to have a performance close to that of the more complex algorithms in most scenarios. The most computationally complex algorithm investigated (SSS) should however be considered in certain settings, especially when packet loss fractions are high and the rates carried by each channel are uneven.

[3]One should notice that the number of loss protection classes in the multichannel case is LC. The division into channels naturally allows for different protection levels per channel even though the intra-channel number of protection classes may be low ($L \to 1$)

Part B

Inciting congestion control through video quality

In this part of the thesis, new solutions addressing the potential problem of misbehaving users in congestion control for video communication are presented. The work is presented in two parts; chapter 4 considers the case of unicast communication while chapter 5 considers the multicast case.

The main ideas and results of this part of the thesis have been published as follows:

Stian Johansen, Anna Kim and Andrew Perkis — "Quality Incentive Based Congestion Control for Multimedia Communication over IP networks". In *Journal of Zhejiang University (Science A)*, Volume 7, Suppl. 1, pp. 7-12, 2006. Presented at Packet Video Workshop, Hangzhou, China, April 2006.

Stian Johansen, Anna Kim and Andrew Perkis — "Quality Incentive Assisted Congestion Control for Receiver-Driven Multicast". In *Proceedings of IEEE International Conference on Communications (ICC-07)*, Glasgow, Scotland, June 2007.

Chapter 4

Inciting congestion control through video quality: Unicast

In this chapter, the potential problem of misbehaving users in the context of congestion control for *unicast* video communication is addressed. The main novelty and idea is to introduce an *incentive in terms of video quality* for adhering to the fair rate sharing mechanisms of congestion control. This is in contrast to the current situation, where there normally is a gain in visual quality to be seen from *not* adhering to congestion control. Earlier approaches for avoiding such misbehavior by receivers have been largely based on changes to protocols, and have not considered visual quality. The presented framework resides at the application layer (source and channel coding), and does not require any protocol changes. It is based on joint source-channel coding (JSCC), and utilizes the algorithms of chapter 3 as an example realization. Model-based simulations are given, along with a discussion of the strengths and limitations of the proposed approach.

4.1 Introduction

As the brief review in chapter 2.3 indicated, congestion control for UDP-based multimedia communication is important for maintaining network stability and for achieving fair sharing of available network resources. As IPTV and similar high-bandwidth services utilizing video streaming are deployed, a lack of proper congestion control can be potentially harmful to the communication of other data (i.e. reliable data transfer using TCP). Perhaps even more critical is video streaming in mobile ad hoc networks (MANETs), in which available transmission

rates typically are much lower than in wired networks. MANETs also exhibit both congestion induced packet loss and packet losses inherent to the wireless medium when multiple video streaming sessions compete for bandwidth in a distributed fashion.

As has been pointed out in literature (see e.g. [18] [19]), current congestion control approaches have a flaw in that they are susceptible to misbehaving entities in the network.

On the source side, the sender may choose to either disregard congestion control entirely, or possibly change the parameters or implementation of congestion control algorithms in order to exceed its 'fair' share of bandwidth. This potential problem has been proposed solved by introducing some form of monitoring by edge routers, whereby the traffic of misbehaving senders is either denied from entering the network entirely or subject to traffic shaping.

The attention in this work is, however, on *bandwidth-greedy receivers*. As pointed out in [19] [17], congestion control schemes that rely on feedback from the receiver side are susceptible to misbehaving receivers. These receivers may intentionally report incorrect information about throughput, packet losses, delays, acknowledgments and other parameters that are used by the sender-side congestion control algorithm to adjust the sending rate.

A side-effect of faulty or intentionally non-conformant congestion control by a subset of senders/receivers may be increased packet losses. Known techniques for packet loss protection (of which the techniques of part A of this thesis are an example) can of course be used to protect the sent data. Thus, packet losses experienced as a result of non-conformant congestion control do not necessarily imply a severely reduced decoded quality at the receiver. In the extreme case, a sender may choose to transmit large amounts of redundancy (packet loss protection) in order to virtually guarantee error-free decoding, without regard to what impact this may have on the other (presumably congestion control conformant) senders/receivers in the network. Clearly, mechanisms need to be in place to avoid such situations.

In this work, sender-side joint source-channel coding (JSCC) techniques are introduced to counteract misbehaving receivers for the case of UDP-based video communication. By realizing that the motivation for non-conformant behavior by a receiver is to gain an improved video quality as a result of the unfair bandwidth share, the aim of this work is to remove this incentive. Specifically, it is attempted to ensure that increased throughput as a result of incorrect feedback reports results in a *degradation* of decoded video quality rather than an improvement. This is done through carefully matching the error resilience characteristics of the video stream to the properties of congestion control algorithms.

4.1.1 Related work

In [17], the authors consider the problem of misbehaving receivers in reliable data transfer under TCP congestion control. By exploiting vulnerabilities in the ACK mechanism from which TCP congestion control adjusts its sending rate, the authors define three different types of misbehavior. Specifically, a user may deviate from the ACK scheme in order to gain an unfairly high throughput. Similar to the motivation and goal of the work presented in this part of the thesis, the authors seek to eliminate the incentive for non-conformant behavior by receivers. Certain changes to the TCP message exchange are proposed that will ideally yield the achieved throughput to be reduced rather than increased if a receiver attempts to deviate from the "normal" ACK scheme.

For the case of TFRC, the protocol specification itself [16] mentions the possibilities for a greedy receiver to communicate incorrect information on the return path to the sender. A specific potential for protocol exploitation is that a receiver may claim to have received packets that were in fact lost due to congestion. The protocol specification document does not consider solutions to this problem. That is however done in [18], where a variant dubbed RTFRC (Robust TFRC) is proposed. RTFRC changes the architecture of TFRC by switching the calculation of the loss event rate from the receiver to the sender, as well as introducing a nonce for verification of feedback messages.

Although not adressing the potential problem of receiver misbehavior, the work of Puri et. al. in [101] deserves a mention in this context. The authors propose a framework based on MD-UXP (see section 2.2.2) and a new proposed congestion control algorithm. The proposed algorithm is dubbed LIMD/H (Linear Increase, Multiplicative Decrease with History), and attempts to smooth the typical highly variable rate of TCP (as depicted in figure 2.11) in order to make it more suitable for continuous media communication. The integration of MD-UXP techniques with congestion control is similar to the work presented in this chapter. However, the framework presented here does not suggest new congestion control algorithms, but rather consider integration with existing congestion control algorithms in order to remove the incentives for misbehavior.

4.1.2 Chapter outline

The work in this chapter is structured as follows. Section 4.2 describes the main aim of this work and provides a general rate-distortion framework formalizing it. Insights into how the approach can be integrated with existing congestion control schemes is given in section 4.3, followed by an example JSCC-based realization in section 4.4. Simulation results are provided in section 4.5, whereafter the chapter is concluded in section 4.6.

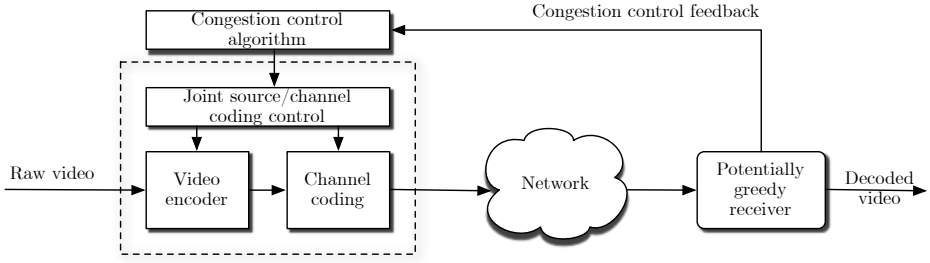


Figure 4.1: General system overview for the unicast case.

4.2 Problem formulation

The related approaches described in section 4.1.1 tackle the problem of misbehaving receivers from an architectural and protocol-centric viewpoint, and do not consider the actual content (in this case video) that is communicated. The approach taken in this work is different in that *congestion-dependent visual quality is used directly as the means of discouraging non-conformant behavior*. By utilizing knowledge of video rate-distortion characteristics, congestion control algorithms and error resilience tools, a joint source-channel coding framework that replaces the existing incentives for misbehavior with incentives for adhering to fair rate sharing is developed.

The key point is the following: Out of *all* possible requested/subscribed rates by a receiver, the *specific* rate yielding the lowest video distortion should be as close as possible to the 'fair' rate indicated by the congestion control scheme in use. Consequentially, a receiver *not* implementing proper congestion control or actively misleading congestion control schemes (in order to attain a higher reception rate) should experience an *increased* visual distortion. Of course, receiving a rate lower than the fair rate will in any case yield a suboptimal distortion.

Formally, the aim is to achieve a congestion-dependent end-to-end[1] video distortion that obeys the following:

$$D(R,\gamma) \;<\; D(R',\gamma) \;\; \forall \;\; R' \neq R, \;\; (R, R' \in \mathbb{R}^{+}) \tag{4.1}$$

where

$$R = C(\gamma). \tag{4.2}$$

Here, $D(R,\gamma)$ denotes the end-to-end distortion D as a function of total transmission rate R and channel conditions (network state) γ. The function $C(\gamma)$

[1]The *end-to-end video distortion D* signifies the distortion as measured between the original, uncompressed video and the resulting decoded video at the receiver.

signifies the congestion control algorithm which determines the "appropriate" transmission rate R based on the network state given by γ.

Figure 4.1 shows the basic system model of the unicast video delivery system considered in this work. The congestion control algorithm acts upon feedback from the receiver. This receiver may or may not be attempting to tamper with feedback parameters in order to achieve an unfairly high throughput. The output of the congestion control algorithm is the 'fair' rate of equation 4.2, and the task of the joint source/channel coding system within the dashed box in the figure is to achieve a distortion-rate performance indicated by equation 4.1.

The intended characteristics of the framework are further visualized in figure 4.2. Here, it is useful to consider the left side of the figure as relating to equation 4.1, while the right side of the figure relates to equation 4.2. The left hand side of the figure shows example distortions for two different transmission rates[2] as a function of congestion. These two rates (R_1 and R_2) relate to the right hand side of the figure, where they are found as the "fair" rates corresponding to two different congestion states (γ_1 and γ_2). Then, according to equation 4.1, rate R_1 should be the rate yielding the lowest distortion at congestion state γ_1, while rate R_2 should yield lowest distortion at congestion state γ_2. This is visualized in the figure. As is evident from the figure, a receiver misleading the sender to transmit a video stream of rate R_1 when R_2 is the fair rate will experience a *higher* distortion than what would have been the case for receiving (the lower) rate R_2. It is mentioned that the distortions D_1 and D_2 corresponding to the non-congested ($\gamma = 0$) case are given by source coder performance.

In this context it is important to note that, in congestion controlled unicast, the receiver normally does not request a specific transmission rate. Rather, the appropriate transmission rate is found by the congestion control algorithm (at the sender side) based on certain congestion related parameters that are fed back from the receiver. Thus, these are the parameters that a receiver may falsify in order to gain an unfairly high bandwidth share. The specific parameters and their importance may vary between different congestion control algorithms.

4.3 Integration with congestion control schemes

This section investigates how the proposed framework can be integrated with relevant existing congestion control approaches. Specifically, the cases of TRFC and AIPD are considered.

[2]It is mentioned that the number of different transmission rates is theoretically infinite, but for visualization purposes only two are shown in the figure.

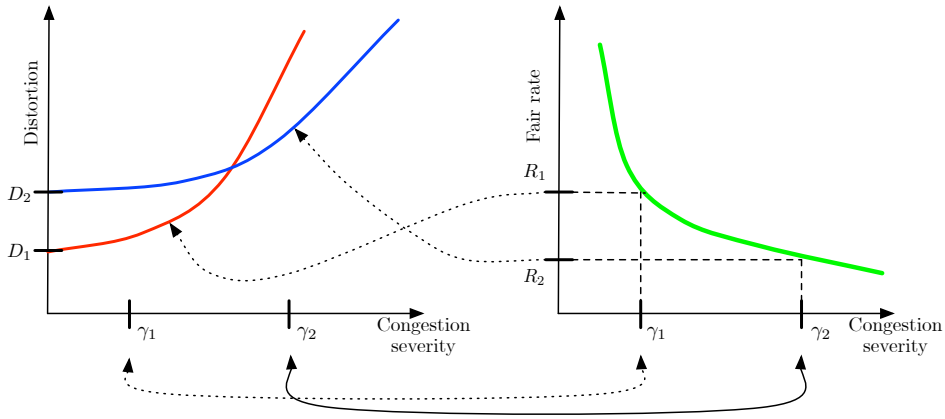


Figure 4.2: Conceptual relationship between video distortion, congestion state and fair rate shares in the proposed framework.

4.3.1 TFRC

In TFRC [16] (see section 2.3.1 for details), a receiver may report a lower loss event rate (corresponding to a lower packet loss fraction) than the real observed value. The result of this type of behavior is shown in [18] to potentially yield a throughput several times the fair rate for the misbehaving receiver.

$$X = \frac{s}{RTT\sqrt{\frac{2bp_{\text{ler}}}{3}} + (t_{\text{RTO}}(3\sqrt{\frac{3bp_{\text{ler}}}{8}})p_{\text{ler}}(1 + 32p_{\text{ler}}^2))} \tag{4.3}$$

Packet loss fraction vs. loss event rate

Equation 4.3 shows the relationship upon which the rate adaption of TFRC is based. As mentioned in section 2.3.1, TFRC uses the 'loss event rate' p_{ler} as a controlling parameter, rather than the packet loss fraction p. A 'loss event' is defined as one or more packet losses within a single RTT. Obviously, loss events do not relate to decoded video quality as directly as the packet loss fraction does, since multiple packet losses may be 'masked' by a single loss event.

In [73], it is shown that the relationship between loss event rate p_{ler} and packet loss loss fraction p can be approximated as

$$p_{\text{ler}} = \frac{p}{1 + \sqrt{\frac{3p}{2}}}. \tag{4.4}$$

Using this approximation in equation 4.3, the required relationship between congestion state and distortion is obtained.

RTT

According to the specification [16], an estimate of the RTT is calculated at the sender based on feedback from the receiver (specifically, using the measured parameters `t_recvdata` and `t_delay`). A greedy receiver can intentionally report these parameters erroneously in order to gain an unfair bandwidth share. For reasons explained in further detail below, the framework presented here is not well suited to deal with such a delay-oriented type of misbehavior. However, the protocol modification for protecting the RTT calculation given in [18] can be combined with the framework proposed here in this case.

4.3.2 AIPD

The "Additive Increase, loss-Proportional Decrease" (AIPD) scheme described in section 2.3.1 reacts directly to packet losses experienced by the receiver. Specifically, the transmission rate decrease is proportional to the experienced packet loss fraction during congestion (equation 4.6), while it is additively increased in the absence of congestion (equation 4.5).

$$X_{i+1} \leftarrow X_i + \alpha \quad \text{(increase)} \tag{4.5}$$

$$X_{i+1} \leftarrow X_i(1 - \gamma p) \quad \text{(decrease)} \tag{4.6}$$

Consequentially, there is no need for an approximation such as in the TFRC case, since packet loss fraction is used directly as the main parameter in congestion control.

4.3.3 Others

While only the mentioned TFRC and AIPD congestion control approaches are simulated in the results section for the unicast case presented here, the framework can be used together with other congestion control schemes. However, the framework here does not lend itself to all congestion control schemes. Specifically, when the controlling parameter(s) γ in the congestion control algorithm do not directly influence video quality, the integration is difficult. An example of such a case (and relating to TFRC, as mentioned above) is when RTT is used for controlling the fair sending rate. This is problematic as there is no direct relationship between delay and distortion (unless packets arrive too late and have to be discarded due to playout deadlines).

It is noted that the TCP-like AIMD family of congestion control schemes (see section 2.3.1) can theoretically be used together with the the presented framework. In practice, this is however unlikely to yield satisfactory performance. This type of schemes, like TCP, typically throttle back sending rate relatively heavily upon detection of even a single lost packet. A single lost packet will in most cases not have a significant influence on video quality (at least for

recent video coding techniques). Thus, a bandwidth greedy receiver would be able to refrain from reporting the loss of a packet without experiencing a severe degradation of video quality (as long as packet losses do not persist). Because of this and the fact that AIMD approaches are inherently ill-suited for communicative video, these approaches are not considered further here. The reader is referred to [102] for further details and simulation results for AIMD.

4.4 Integration and optimization

Realizing the framework deals with obtaining a rate-distortion performance of the video stream that is in keeping with the characteristics defined by equations 4.1 and 4.2. In other words, the implementation should *avoid the case where an unfairly high rate yields a lower distortion than the fair rate.* To this end, the following simple generic algorithm is proposed:

Algorithm 5 Joint video quality optimization and congestion control incitement

Require: Network state γ as experienced/reported by the receiver

1: Calculate fair transmission rate R from equation 4.2, for the congestion control scheme in question
2: Perform rate-distortion optimization that minimizes distortion D at the rate R given network state γ

A requirement for the above algorithm to work as intended is that distortion D increases when congestion gets increasingly severe. When the congestion state and, correspondingly, the fair rate (through equation 4.2) is quantified via the intensity of packet losses, this certainly holds. It may however not always be the case, a point which is strongly related to that made above regarding e.g. RTT-based congestion control approaches.

To summarize, the optimization algorithm takes the following into account:

- Source coder characteristics (rate-distortion performance)
- Error resilience characteristics (e.g. the use of channel codes)
- Congestion control characteristics (relationship between congestion state and the fair rate per user)

Algorithm 5 simply aims to do the following: For any given rate R, minimize distortion D assuming the corresponding congestion state γ. R and γ are, as mentioned, related through equation 4.2 as depicted on the right hand side of figure 4.2. As will be shown in simulation results, the optimization yields the following:

- Minimization of distortion for well-behaved receivers

Parameter	*Details*
RTT	The round-trip time (in seconds)
t_{RTO}	TCP retransmission timeout value (in seconds)
p_{ler}	The loss event rate
b	Number of packets acknowledged by a single ACK packet

Table 4.1: TFRC parameters.

- Approximating the performance inidicated by equation 4.1, thus introducing an incentive in terms of distortion for potentially misbehaving receivers to adhere to fair rate sharing schemes.

As an example implementation of the optimization task, the packet loss protection algorithms of chapter 3 are used. Note however that the algorithms are used in single-channel mode here, and are thus largely equivalent to the original algorithms in [87] and [90]. As the optimization algorithm simply performs rate-distortion optimization considering (taking into account congestion control characteristics), the proposed framework does not increase complexity. Thus, real-time operation is assured; an obviously critical feature of any system dealing with congestion control.

Selected simulation results are given in the following.

4.5 Simulation details and results

This section provides a set of model-based simulations to validate the performance of the framework. The specific optimization algorithms, source models and channel models are identical to those of sections 3.2.2, 3.4.1 and 3.4.2, respectively.

The simulations aim to visualize the intended performance of the proposed framework, and to quantify how accurate the characteristics of equation 4.1 can be approximated. The above described TFRC and AIPD congestion control schemes are simulated under a selection of transmission rates, network states and parameter settings.

4.5.1 TFRC

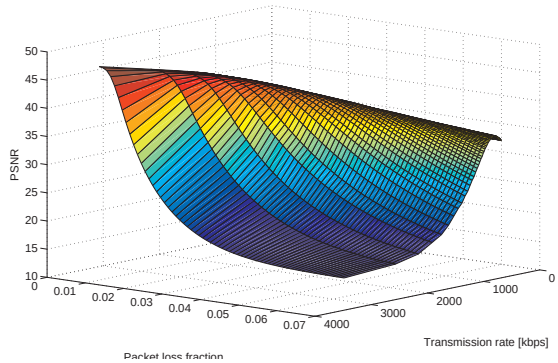
The controlling parameters of the TFRC algorithm are listed in table 4.1. As described in section 4.3.1, the approximation between packet loss ratio and loss event rate given by equation 4.4 is employed. In accordance with [18], the TCP retransmission timeout t_{RTO} is set to $2.5RTT$. Also, for all simulations shown, the number of packets b acknowledged by a single ACK packet is set to one. The SSS algorithm of section 3.2.2 was used for doing optimization for all the shown plots.

The influence of the packet loss distribution parameter ρ in the Gilbert channel model is shown in figures 4.3 and 4.4. Figure 4.3 shows the PSNR performance of the system as a function of packet loss fraction and transmission rate. Keeping in mind the intended performance of the resulting system as described through equations 4.1 and 4.2, the plots should be read as follows: Given a channel state (here described by the packet loss fraction), the point of optimal PSNR in the plot should occur at the rate R as given by equation 4.2 (here given by the TFRC throughput equation). In order to visualize this more clearly, figure 4.4 shows the points where optimal PSNR is attained (dashed lines) together with the "intended" optimality curve (the fair rate as calculated by the TFRC throughput equation).

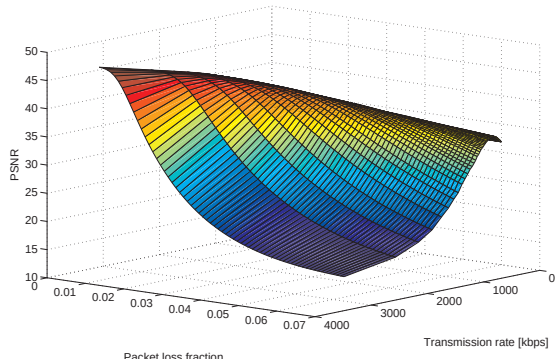
As can be seen from the figures, the gap between the attained optimality curve and the intended optimality curve is low for small values of the ρ parameter and increasing for increasing ρ. This is explained by the fact that (as noted in section 3.4.2), $\rho = 0$ corresponds to a binomial distribution while an increasing ρ corresponds to increasing uncertainty about the number of packet losses to be experienced ($p_N(n)$ tending towards a uniform distribution). Thus, the optimization algorithm will increasingly take into account the possibility of experiencing above-average packet losses as ρ increases. This yields a loss protection (and corresponding distortion-loss performance) skewed towards an above-average number of lost packets.

Figure 4.5 shows the influence of RTT on the ability of the system to align the intended points of optimality and the resulting optimality points (i.e. experiencing maximum PSNR at the fair rates). The three plots show the situation for RTT values of 50, 200 and 300 milliseconds. Results show that the accuracy of tracking the intended curve decreases as the RTT increases. This effect is closely related to the fact that an increasing RTT will lead to a lower fair transmission rate (according to the TFRC throughput equation). The lower transmission rates translate to a lower number of packets and hence fewer degrees of freedom for the optimization algorithm to adjust the error resilience properties of the stream. This is particularly noticeable at the lowest packet loss fractions (see e.g. plot (c)).

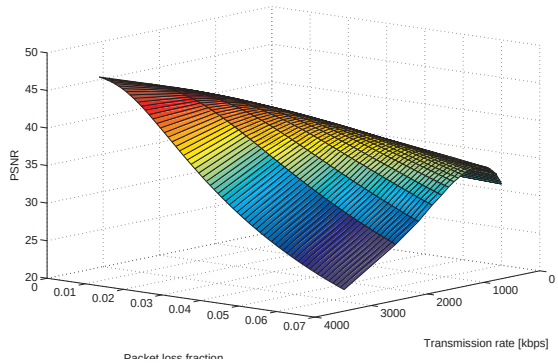
The performance of the system for different packet sizes is shown in figure 4.6. The results show that the relative performance is comparable for all packet sizes (250, 750 and 1500 byte packets are shown in the figure). Notice that, as the packet size influences the fair rate of TFRC, the transmission rate values on the y-axis varies between the figures. Also notice that, for the lowest transmission rate (for packet length 250 bytes in plot (a)), performance is noticeably better than for the lowest rate case in figure 4.5. This is due to the fact that, even though the transmission rates are comparable, the smaller packet size allows for more degrees of freedom in the allocation of channel codes. Hence, a closer alignment of the two curves is attained.



(a)

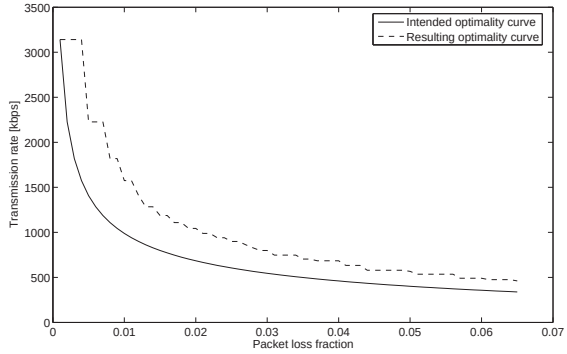


(b)

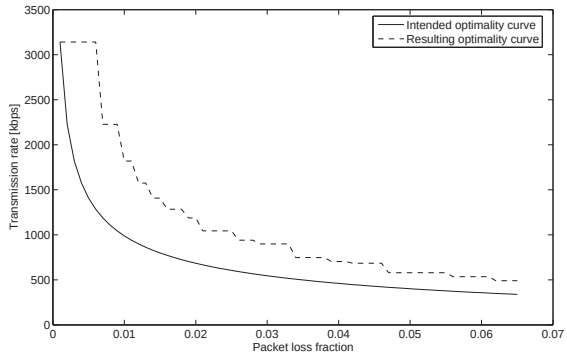


(c)

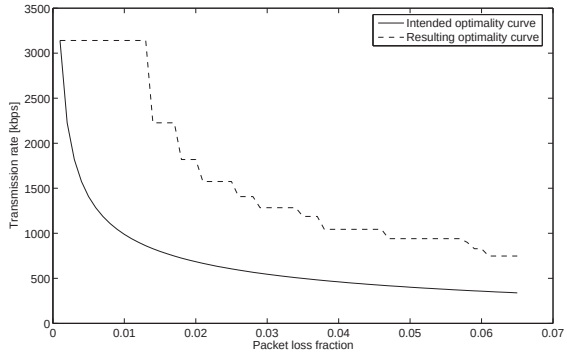
Figure 4.3: Resulting PSNR performance as a function of packet loss fraction and transmission rate for the Gilbert channel model parameter ρ equal to 0.01 (a), 0.1 (b) and 0.5 (c). Packet length is kept constant at 1000 bytes for all plots, and using an RTT of 100ms.



(a)

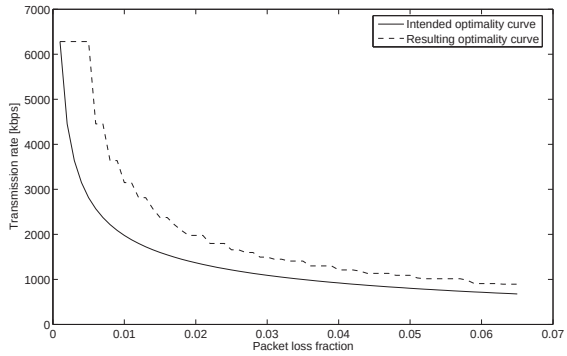


(b)

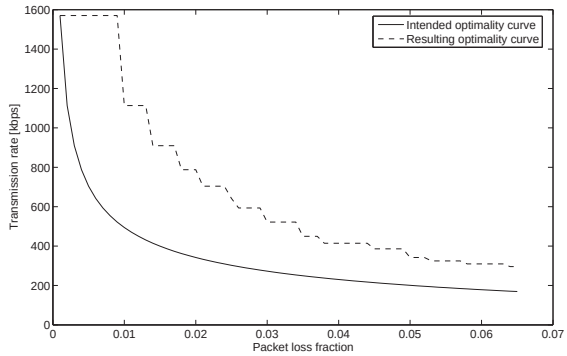


(c)

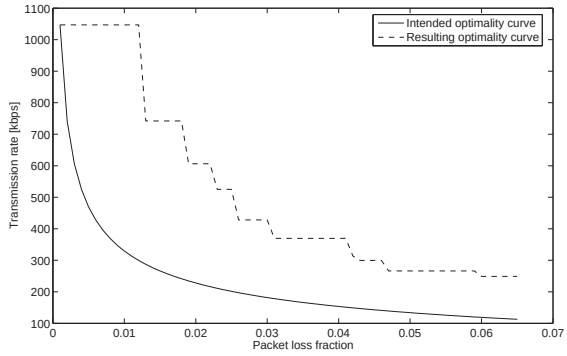
Figure 4.4: The fair rate R (solid line) and the actual points of maximal PSNR (dashed line) in figure 4.3. Shown for Gilbert channel model ρ parameter equal to 0.01 (a), 0.1 (b) and 0.5 (c). Packet length is kept constant at 1000 bytes for all plots, and using an RTT of 100ms.



(a)

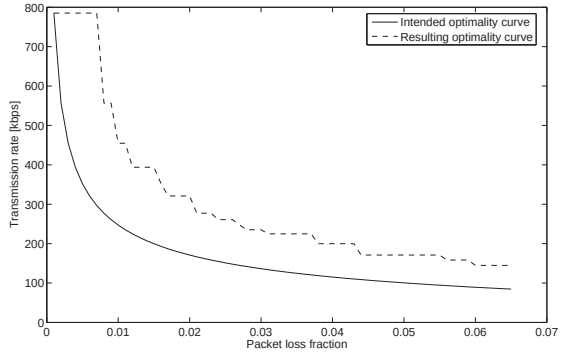


(b)

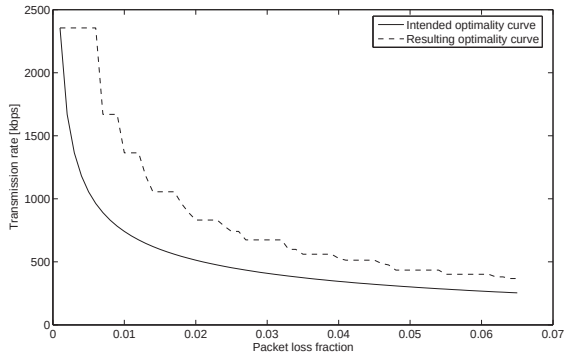


(c)

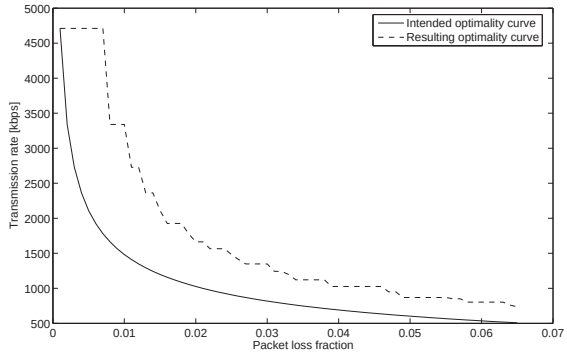
Figure 4.5: The fair rate R (solid line) and the actual points of maximal PSNR (dashed line). Shown for RTT equal to 50ms (a), 200ms (b) and 300ms (c). Packet length is kept constant at 1000 bytes for all plots, with the Gilbert channel parameter $\rho = 0.1$.



(a)



(b)



(c)

Figure 4.6: The fair rate R (solid line) and the actual points of maximal PSNR (dashed line). Shown for packet sizes 250 bytes (a), 750 bytes (b) and 1500 bytes (c). RTT is kept constant at 100ms for all plots, with the Gilbert channel parameter $\rho = 0.1$.

4.5.2 AIPD

For the case of AIPD, we here consider the effects of varying the channel model parameter ρ and the packet size. Referring to the AIPD rate decrease strategy (see equation 2.9, simulations are shown for the proportionality constant $\beta = 10$. The initial rate from which the decrease is calculated is 1200 kbps for all plots.

Figures 4.7 and 4.8 show the performance attained for ρ equal to 0.01, 0.3 and 0.6. The way of interpreting the plots is the same as in the TFRC case. The same tendency that was seen for the TFRC case is recognized in these results; the accuracy in aligning the intended points of optimality is best for a low value of ρ and decreasing as ρ increases. The reasons for this are obviously the same is in the case of TFRC, since this does not depend on the congestion control scheme used but rather the channel characteristics and the optimization algorithm.

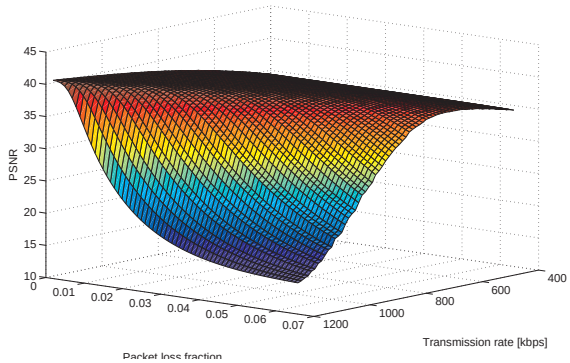
Figure 4.9 shows how different packet sizes impact the performance of the system. The effect of having a lower number of degrees of freedom in the allocation of channel codes becomes pronounced at lower rates and large packet sizes, as can be seen from plot (c). For smaller packet sizes (see plot (a)) where the degrees of freedom for the optimization algorithm is larger, the accuracy in aligning the intended and obtained optimality points is good.

It is noted that the case of AIPD is slightly different from TFRC in that it has two different modes of operation, namely the *increase* and *decrease* phases (see equation 2.9), of which the decrease phase is considered in the results shown here. It is however mentioned that the framework presented works for the increase phase as well. In the increase phase there should be no congestion (i.e. zero packet loss), and consequentially there will be no (or at least minimal) error protection added. A receiver falsely reporting zero packet loss when the network actually *is* congested will therefore achieve a very low PSNR (if being able to decode anything at all).

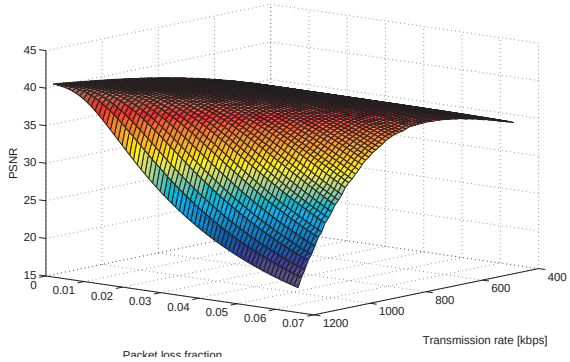
4.5.3 Discussion

The presented results indicate that the presented framework is able to remove (or at least strongly limit) the incentive for receivers to falsify feedback information in order to gain an unfairly high throughput. Since the optimization algorithm used is simply minimizing distortion for a receiver that does not attempt to falsify feedback information, it has the fortunate side-effect of giving an optimized visual quality for well-behaved receivers.

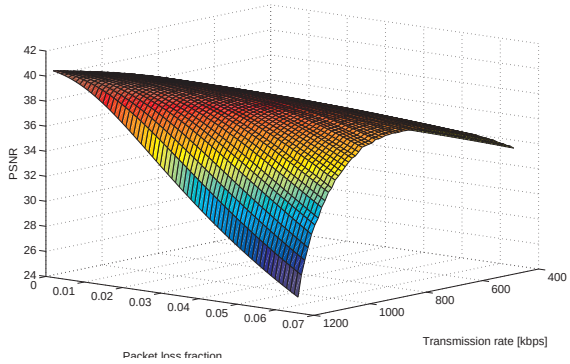
The results show that there is a slight difficulty in aligning the points where maximum PSNR performance is obtained with the points implied by equations 4.1 and 4.2 for certain combinations of parameter settings and network states. Specifically, it is seen that the combination of higher values of the ρ parameter and larger packet sizes may give a considerable deviation from the desired characteristics. This could be improved by adding constraints to the



(a)

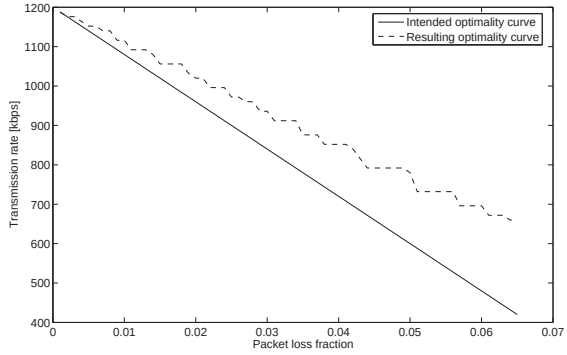


(b)

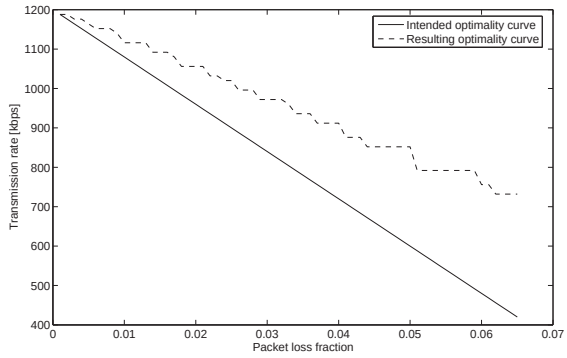


(c)

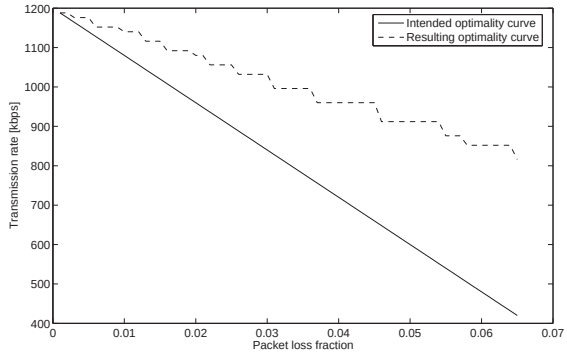
Figure 4.7: Resulting PSNR performance as a function of packet loss fraction p and transmission rate R for the Gilbert channel model parameter ρ equal to 0.01 (a), 0.3 (b) and 0.6 (c). Packet length is kept constant at 500 bytes for all plots.



(a)

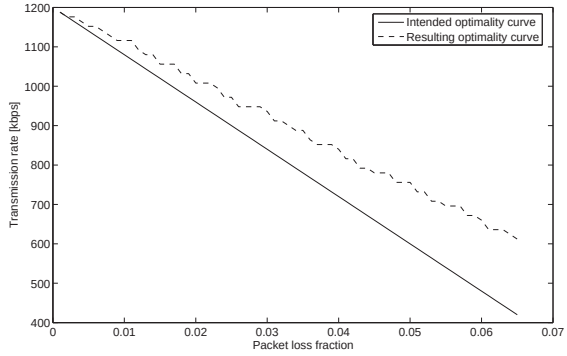


(b)

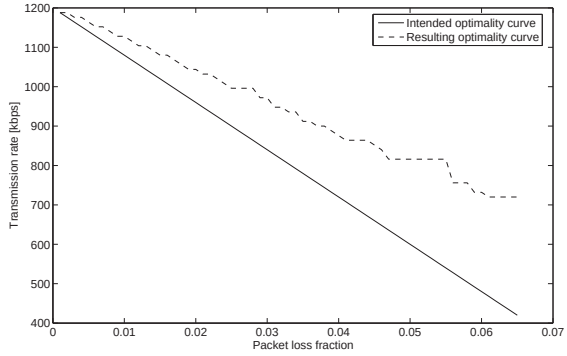


(c)

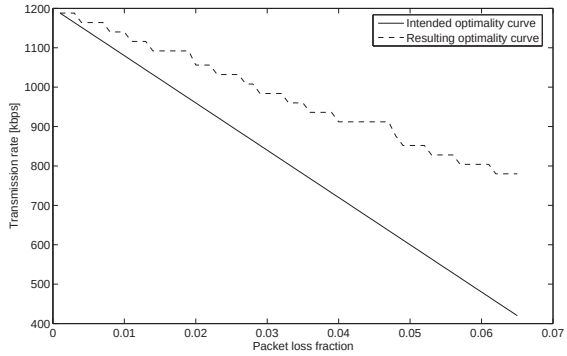
Figure 4.8: The fair rate R (solid line) and the actual points of maximal PSNR (dashed line) in figure 4.3. Shown for Gilbert channel model ρ parameter equal to 0.01 (a), 0.3 (b) and 0.6 (c). Packet length is kept constant at 500 bytes for all plots.



(a)



(b)



(c)

Figure 4.9: The fair rate R (solid line) and the actual points of maximal PSNR (dashed line). Shown for packet sizes 250 bytes (a), 750 bytes (b) and 1500 bytes (c). The Gilbert channel parameter $\rho = 0.1$.

optimization algorithm (e.g. its dependence on the packet loss pdf), or by performing the optimization for all transmission rates and then adjusting the allocation of channel codes to better match the intended maximum-PSNR points. This would, however, give a suboptimal distortion for well-behaved receivers. As it is reasonable to assume that the majority of receivers are well-behaved, algorithm modifications that incur a performance penalty for these should be avoided.

4.6 Conclusions and future work

This chapter has presented a framework for inciting congestion control for unicast video communication. Through an integration of equation-based congestion control and joint source-channel coding, the system yields a congestion-dependent visual quality that removes the incentive for receivers to not adhere to congestion control strategies. Specifically, the proposed framework will yield a *decreased* rather than increased visual quality for a receiver that does not implement or actively attempts to falsify congestion feedback in order to gain an inflated bandwidth share.

Recommendations for future work includes:

- Trials in real networks. In order to assess how the dynamics of larger networks influence the incentives for fair rate sharing, trials in actual networks are needed. An important performance measure that needs further investigation in this context is the *time scale* over which a misbehaving receiver will experience reduced video quality. This will depend on both the volume of and type of competing traffic.

- Integration with IETF RTP payload format. The draft IETF standard in [103] specifies an RTP payload format for MD-UXP type packet loss protected multimedia streams. The integration of the proposed approach with this payload format should be investigated, along with any restrictions this may pose and their eventual implications.

Chapter 5

Inciting congestion control through video quality: Multicast

The underlying rationale of the approach investigated in chapter 4 is here extended to the case of video delivery using multicast. As will be shown, the differences between unicast and multicast calls for a separate treatment where the specifics of multicast delivery in general, and layered video specifically, are considered. Similar to chapter 4, the main novelty is *the utilization of video quality as a means of avoiding receiver misbehavior*. Additionally, a novel *layering scheme* is proposed. As in chapter 4, the problem is proposed solved through the integration of joint source-channel coding (JSCC) techniques with congestion control techniques for receiver-driven layered video multicast. Simulation results highlight the achieved and intended performance of the resulting system.

5.1 Introduction

The potential problem of misbehaving receivers in multicast video delivery and communication is significantly different to the unicast case. As one of the main uses of multicast is large-scale delivery, most approaches refrain from using feedback from receivers to source(s) due to the 'feedback implosion' problem. As described in section 2.3.2, receiver-driven layered multicast using layered video coding has been studied extensively for the delivery of streaming video due to its distributed rate-adaptivity features and inherent bandwidth efficiency. In these approaches, receivers subscribe to the appropriate subset of the available video layers based on an adaptation algorithm/protocol (see examples in section 2.3.2). Obviously, a greedy receiver may subscribe to a larger number of layers than what

is fair from a congestion control point of view. In this chapter, a novel framework for avoiding this is proposed. Similar to the unicast case, it is attempted to introduce an incentive for proper rate adaptation by receivers through closely relating video quality to the level of congestion in the network.

5.1.1 Related work

The outlined problem has been studied by Gorinsky et. al. in [19] [104]. The approach taken by the authors is to implement mechanisms to *prevent* misbehavior rather than *discourage* it. Specifically, a two-part countermeasure is proposed; congestion-dependent access to subscribe to groups ("SIGMA", Secure Internet Group Management Architecture) together with a system for distributing keys to receivers ("DELTA", Distribution of ELigibility to Access). Congestion dependent keys are distributed to both receivers and edge routers, allowing for architecture-based prevention of misbehavior by receivers. The key-based access attempts to ensure that receivers can only subscribe to groups that correspond to a fair share of the bandwidth.

A related approach is taken in [105], where the authors use MINC (Multicast-based Inference of Network internal Characteristics) to calculate the loss rates of individual links in a multicast tree. This can then be used to detect misbehaving receivers, as the links involved in relaying multicast data to these receivers will tend to experience a higher fraction of lost packets. In order to make this reliable, the authors investigate how MINC techniques can be made robust, as they are also inherently fragile towards receiver misbehavior (through falsified feedback). For receiver-driven multicast, the authors go on to propose a key distribution scheme similar to the DELTA approach mentioned above. Thus, this solution is architecture-based similar to [19]. There are also some open questions regarding the scalability of this approach, as MINC can involve significant volumes of feedback data to the receiver; this is potentially harmful to large-scale deployment.

The approach proposed in this chapter is different from the approaches described above in that it is not based on changing protocols or architectural-level changes. Rather, it relies on integrating congestion control mechanisms with source and channel coding. The approach proposed here does not require any feedback from receivers to the sources, and should thus be highly scalable to large-scale multicast delivery of video data.

5.1.2 Chapter outline

The work in this chapter is structured as follows. Section 5.2 reworks the formulation given for the unicast case, taking into account the characteristics of multicast. Section 5.3 gives an example realization of the framework where the multichannel optimization algorithms of chapter 3 are used. Some relevant

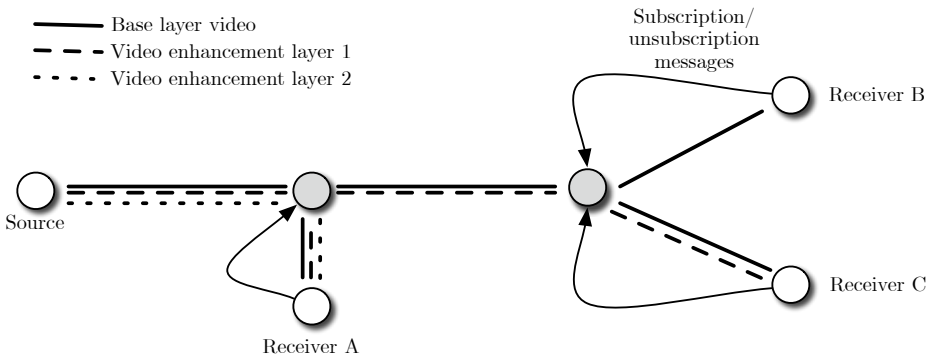


Figure 5.1: Example receiver-driven layered video multicast scenario.

issues are discussed in section 5.4, followed by remarks on integration with known receiver-driven multicast schemes in section 5.5. Simulation results are given in section 5.6 and concluding remarks then follow in section 5.7.

5.2 Problem formulation

The potential problem of oversubscribing receivers in receiver-driven multicast is addressed. The presented framework offers a set of subscription alternatives for receivers in which a penalty in terms of visual quality will be experienced by oversubscribing receivers. Thus, as in the unicast case, the incentive for oversubscription that exists in current systems is greatly reduced.

Figure 5.1 show the basic system model for multicast delivery of layered video. Referring to the overview given in section 2.3.2, the case of receiver-driven layered multicast is considered. In such systems, each video layer is normally communicated in a single multicast group. Receivers then subscribe/unsubscribe multicast groups (and thereby video layers) in order to adjust their receiving rate.

The approach presented here is based on adjusting the erasure resilience properties of the possible group subscription combinations such that each is tailored for the network state at which it should be the prime alternative for subscription by receivers. Specifically, the aim is to "design" the erasure resilience properties of the set of group subscription combinations such that at each specific congestion state there is *exactly one* subscription combination that will give the highest decoded visual quality. Furthermore, this subscription combination should have a total transmission rate that corresponds with the fair rate as given by the congestion control scheme. Thus, this is analogous to the formulation in equations 4.1 and 4.2. However, as the layered multicast case implies a limited number of possible transmission rates, a slight reformulation is needed.

5.2.1 Distortion-congestion relationships

Formally, the intended performance is characterized by

$$D(s,\gamma) \;<\; D(s',\gamma) \quad \forall \;\; s' \neq s \in \mathcal{S} \tag{5.1}$$

$$R_s(s) = C(\gamma), \tag{5.2}$$

where s is a specific subscription combination from the set $\mathcal{S}$ of possible subscription combinations, while network state is denoted by γ. The function $D(\cdot)$ has the same interpretation here as before, namely the end-to-end distortion (distortion in decoded video with respect to the raw video input to the encoder). $R_s(\cdot)$ denotes the rate associated with a particular subscription combination, while $C(\cdot)$ gives the fair transmission rate as determined by the congestion control scheme used.

This can be seen as a many-to-one mapping $\mathcal{M}$ of network states to subscriptions;

$$\mathcal{M} : \Gamma \mapsto \mathcal{S}, \tag{5.3}$$

where Γ is the set of possible network states. Thus, a *specific* network state cannot map to different subscriptions, while several *different* network states can map to the same subscription (due to the steps in rate that are inherent in layered multicast).

A practical interpretation of the above is that when the network state deteriorates (onset of congestion), the receiver should reduce its subscribed rate according to the congestion control scheme. However, given the congested state of the network, there should be a *gain in terms of visual quality* by reducing the rate. Obviously; there should also be a visual quality gain to be found from increasing the receiving rate when congestion is alleviated. Thus, there is no longer a disparity between optimization of visual quality and adhering to the congestion control scheme.

The general concept is visualized in figure 5.2. The figure shows the case of three different subscription alternatives. In line with the described framework, the subscription alternative yielding the lowest distortion at a particular congestion level should be the one corresponding to the fair rate at that specific congestion level. Compared to the unicast case in figure 4.2, as the case is for layered multicast, the number of different rates is quantized into a finite number rather than the theoretically infinite number of different rates in the unicast case. Consequentially, each different subscription alternative corresponds to a *range* of congestion states, as indicated by equation 5.3 above.

5.2.2 Layer structure

As indicated in figure 5.3, the layer hierarchy used in most multicast schemes for video is strictly cumulative. That is, all data contained in subscription level i is

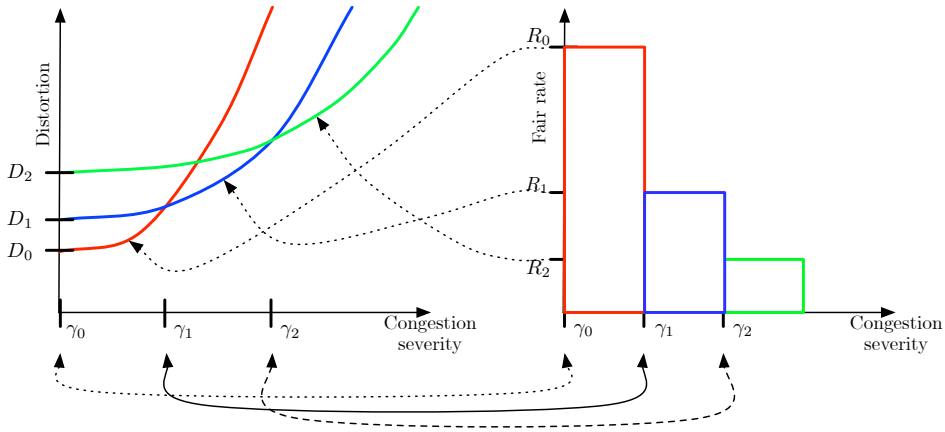


Figure 5.2: Conceptual relationship between video distortion, congestion state and the different possible subscription rates. Shown for the three-layer case.

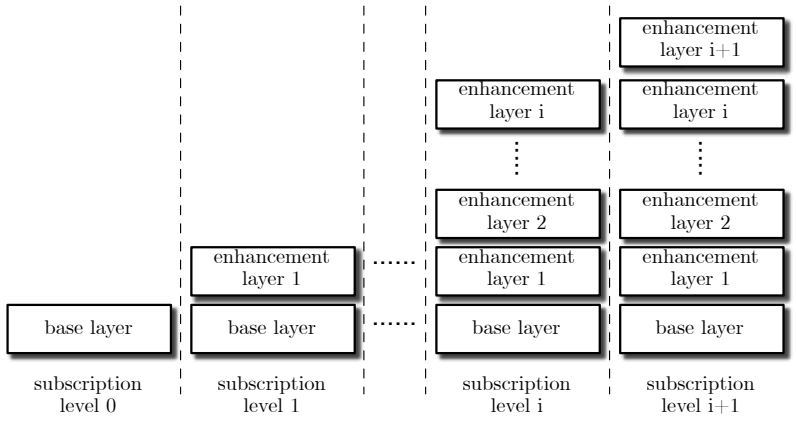


Figure 5.3: Strict cumulative layering structure.

reused as a part of subscription level $i+1$. Certainly, much of the bandwidth efficiency in layered video multicast stems from this layer organization.

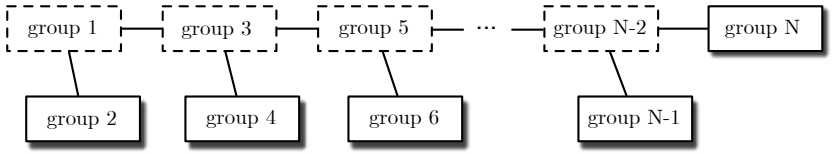
However, using this structure without modifications is not straightforward in the case of the proposed framework. Consider the following, referring to figure 5.3. Given that subscription level i is optimized for network state γ_i, this data should certainly have erasure resilience properties that allow it to be successfully decoded at this particular network state. Now, consider a receiver attempting to oversubscribe, adding a layer and thus requesting subscription level $i+1$. Due to the cumulative property of the layer structure, subscription level i will be included *unaltered* as a subset of subscription layer $i+1$. Now, although the additional layer may not be decodable for this receiver (as it should have erasure resilience properties corresponding to a *less* congested state), this receiver will still be able to decode layers corresponding to subscription level i[1]. Thus, the receiver will not experience an *increase* in video distortion as a result of the oversubscription, as is required by equation 5.1. This is due to the fact that enhancement layer data cannot (on average) reduce the decoded quality, only improve it. As a result, there is no incentive (in terms of visual quality) for the receiver to reduce its subscription to the appropriate level i.

Semi-cumulative layering

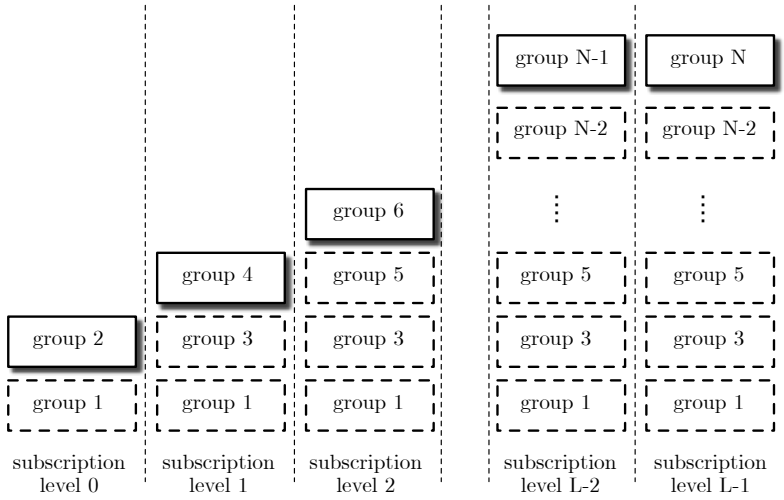
Figure 5.4 shows a different layering structure that can be used for preventing this situation. In the tree structure in figure 5.4(a), the dotted-lined boxes are cumulative (as in the classical case), while the solid-line boxes are not. This is further shown in figure 5.4(b), where the different possible subscription combinations are shown. As is apparent from the figures, each subscription combination must be terminated with one of the leaf nodes (solid-lined boxes) of the binary tree structure in figure 5.4(a).

It is required that each leaf node not only conveys refinement information but also *a fraction of the data needed to decode the base layer* (see figure 5.5). This is the key for encouraging the receiver to reduce its subscription level in the case of congestion. Specifically, by adjusting the erasure protection for this *prefix* of the bitstream (the part of the bitstream between markers '1' and 'b_0' in figure 5.5), it is possible to tune the resulting expected distortions such that the intended mapping of congestion state to visual quality (equation 5.3) is attained. This layering approach gives an inter-group data dependency yielding the following property: *A failure to decode data in the refinement data groups will give an increased distortion compared to what would otherwise result from the lower layers alone.* Data organization within groups is shown in figure 5.5. The figure shows a structure using the unequal packet loss protection schemes of chapter 3, in which

[1]It is here assumed that the packet loss fraction of the congested link does not significantly increase because of the oversubscribed layer, leading to a failure of decoding video layers in subscription level i.



(a) Relationship between multicast groups.



(b) The resulting different subscription alternatives.

Figure 5.4: Proposed alternative layering structure.

the different parts of the bitstream (rows in the figure) are assigned channel codes according to their relative importance for video distortion. It is however mentioned that using the techniques of chapter 3 is an example implementation, others are possible. The key point to be read from figure 5.5 is the inclusion of the bitstream prefix in the leaf multicast group rather than in the lowest (base) layer, as in the strict cumulative layering structure.

Integration with strict cumulative layering

It is noted that the framework can be implemented using the strictly cumulative layering approach as well, if the condition of equation 5.1 is relaxed slightly. Specifically, if the requirement is loosened as follows;

$$D(s,\gamma) \;\leq\; D(s',\gamma) \quad \forall \;\; s' \neq s \in \mathcal{S}, \tag{5.4}$$

then the classical cumulative layering approach can be used. Notice that this slight reformulation implies that an oversubscribing receiver could experience the

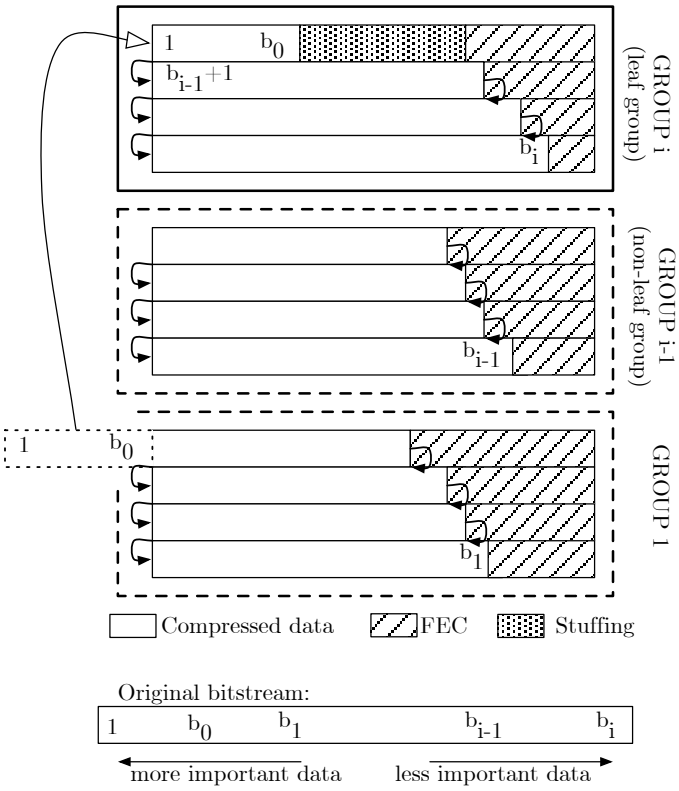


Figure 5.5: Data organization within the multicast groups. As in chapter 3, packets are formed as vertical 'slices' within each group.

same distortion as a well-behaved receiver, *but cannot experience a better (lower) distortion*. Thus, this is a weaker incentive than the initial formulation, where an oversubscribing receiver is required to experience *increased* distortion as a result of oversubscription. This may, however, be a viable alternative if the semi-cumulative layering structure is not desired. Further comments on this are given in section 5.4.

5.3 RD optimization

Based on the multichannel optimization techniques of chapter 3, an optimization algorithm for attaining the performance indicated by equations 5.1 and 5.2 is proposed. Let f_i $(i \in 1, \cdots, N)$ denote the allocation of channel codes to the prefix in leaf node l (see figure 5.5). As defined earlier, $\mathcal{S}$ denotes the set

of possible subscriptions s, and γ denotes network state. In the pseudo-code of algorithm 6 below, it is assumed that the elements s_i of $\mathcal{S}$ are ordered from low to high rate.

Algorithm 6 Algorithm for optimizing error resilience of prefixes

1: Set f_i to the maximum possible protection for all i.
2: **for all** $s_i \in \mathcal{S}$ **do**
3: Find the optimal allocation of channel codes to the multicast groups (video layers) using the multichannel packet loss protection algorithms of chapter 3, optimizing for channel state γ_i.
4: Do not modify allocations for groups found for $j < i$.
5: **end for**
6: **for all** $s_i \in \mathcal{S} \mid i > 1$ **do**
7: **while** $D(s_i, \gamma_i) < D(s_{i-1}, \gamma_{i-1})$ **do**
8: Reduce f_i, the allocation of channel codes to prefix i.
9: **end while**
10: **end for**

The first part of the algorithm (lines 1 through 5) allocates channel codes to the different groups that constitute the different subscriptions, while the latter part of the algorithm (lines 6 through 10) fine-tunes the rate-distortion performance according to equation (5.1). It is important to note that the algorithm first allocates channel codes to the lowest-rate subscription and then progresses to increasingly higher rate subscriptions. Due to the cumulative nature of the non-leaf nodes, channel code allocation to these nodes can not be changed during optimization of higher-rate subscriptions. A consequence of this is that the higher-rate subscriptions will have an erasure resilience property that is (partly) optimized for worse network conditions than the conditions in which they are appropriate for use (in terms of congestion control)[2]. This motivates the second part of the algorithm, where the erasure code allocation to the prefix of the compressed bitstream is adjusted.

5.4 Relevant issues

5.4.1 Bandwidth efficiency

The tree structure from figure 5.4(a) gives a certain bandwidth inefficiency compared to the strictly cumulative case. This is due to the fact that each leaf node in figure 5.4(a) is only used in *one* subscription alternative; they are not re-used as subsets of higher-rate subscriptions as in the fully cumulative case. The bandwidth inefficiency resulting from this can however be kept fairly low.

[2]It is emphasized this systematic rate-distortion inefficiency is not due to the proposed framework but is inherent in layered multicast.

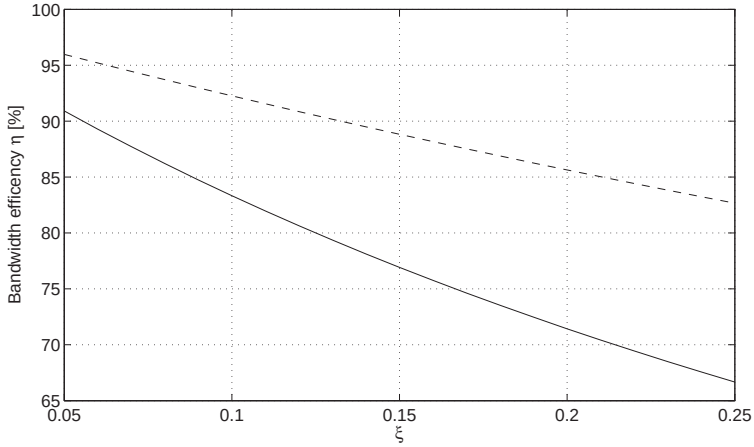


Figure 5.6: Bandwidth efficiency compared to strictly cumulative multicast, as a function of the leaf-group rate parameter ξ. Solid line: Linearly increasing subscription rates. Dashed line: Exponentially increasing subscription rates.

By making the leaf groups (solid line boxes) have a small rate compared to the cumulative groups (dashed-line boxes), bandwidth efficiency can still be close to that of strictly cumulative layered multicast. Let L be the number of different layers (in the case of strictly cumulative layer organization), with rates given by $R_i (i \in \{1, \cdots, L\})$, and let the factor ξ be the fraction of the received rate that is communicated in the leaf multicast group. Assuming that the rates R_i are ordered from low to high subscription level with increasing i, the bandwidth efficiency η of the new layering structure when having N leaf nodes is

$$\eta = \frac{\sum_{i=1}^{N} R_i}{R_1 + \sum_{j=2}^{N} \left(R_j + \xi \sum_{k=1}^{j-1} r_k \right)} \tag{5.5}$$

Figure 5.6 shows η as a function of ξ for the case of $N = 5$. The solid line is for the case where rate increases linearly (i.e. $\mathbf{r} = \{200, 200, 200, 200, 200\}$kbps), while the dashed line is for exponentially increasing rates [3] (i.e. $\mathbf{r} = \{100, 200, 400, 800, 1600\}$ kbps). As is apparent from the figure, bandwidth efficiency can be reasonably close to that of strictly cumulative layered multicast when the ξ factor is low.

[3] It is mentioned that exponentially increasing layer rates is often used to mimic the slow-start phase of TCP.

5.4.2 Address usage

As each multicast group is associated with a single multicast address, the address usage will increase by using the semi-cumulative layering approach. Specifically, using a binary-tree based structure (as shown here) the number of multicast addresses used will be $2N - 1$ when having N different subscription alternatives. Compared to the strictly cumulative layered case (which requires N addresses for the same situation), this represents a marked increase in address usage. However, this may not be prohibitive for low N (as is usually the case in layered video multicast, where N is typically no greater than 4-5).

Note that the number of leaf nodes N will be equal to the number of layers L in the case of a layered video coder (as opposed to the case with a fully embedded coder, where N can be chosen freely).

5.4.3 Possibility of subscribing multiple leaf nodes

The framework as presented so far has a potential weakness in that a oversubscribing receiver may subscribe to multiple leaf node multicast groups. By doing this, a receiver could potentially decode the base layer prefix intended for a lower rate (and thereby being more packet loss resilient), and use this for decoding the leaf node of a higher layer. This can be prevented by including a prefix of the topmost subscribed layer as well as the prefix of the base layer in the leaf node groups. In this way, it is made sure sure that the topmost layer cannot be decoded if packet losses are more severe than what the layer was intended for. Thus, with this modification, the proposed framework cannot be circumvented by a greedy receiver through subscribing multiple leaf nodes.

5.4.4 Temporal considerations

An important consideration is the temporal extent over which the incentive defined through equation 5.1 will be effective. If the traffic of misbehaved users is assumed to make up a minor fraction of the total traffic volume over the congested link(s), the well-behaved traffic will throttle back its rate and eventually alleviate congestion. This will have the effect of gradually removing the incentive introduced here for a misbehaving receiver to reduce its subscription. Two main factors decide the temporal extent of the effectiveness of the incentive:

- *IGMP leave latencies.* The unsubscription of multicast groups is handled by the IGMP protocol (see section 2.3.2). Routers will invoke a polling procedure to test whether a multicast group is still active. If not, the router will unsubscribe it from the next level in the multicast tree. This polling procedure can typically take up to about 10 seconds [72], depending on the size and topology of the multicast tree. During this period, the effect of decreased subscription from well-behaved receivers will not have

an impact on congestion state; thus warranting the effect of the intended distortion penalty for misbehaving users.

- *Sustained increased loss rate.* Simulation results in [104] indicate that misbehaving receivers tend to experience a sustainied increase in the fraction of lost packets even after competing traffic has throttled back (i.e. after IGMP latencies have been accounted for). Although this effect depends on the topology and the nature of competing traffic, this will yield a longer (>100s is reported in [104]) imbalance in congestion state, thus warranting a sustained effect of the intended distortion penalty for a misbehaved user.

5.5 Integration with congestion control schemes

Since the framework presented here is not a congestion control scheme in itself, it is necessary to integrate it with already existing and deployed congestion control schemes. Important points in this respect include:

- *Layer structure.* If the formulation of 5.1 is to be enforced, the tree structured layer organization from figure 5.4(a) must be used. This has certain consequences for the join/leave administration by users. Referring to figure 5.4, it is clear that a subscription increase implies unsubscribing of *one* group and subscribing *two new* groups. Similarly, reducing the subscription level implies unsubscribing two groups and subscribing to one new group. As mentioned in section 5.2.2, the "normal" strict cumulative layering scheme can also be used, in which case the above mentioned issues are not valid.

- *Calculation of fair rate.* It is required that a useful mathematical relationship between network state and fair rate for a single receiver exists, since the proposed framework relies on equation 5.2. This is not to say that the congestion control scheme needs to be equation based, but that its rate control can be approximated mathematically. An example of such a case is given in the following.

5.5.1 Integration with RLC/FLID

The congestion control algorithms RLC [71] and FLID-DL [72] (see section 2.3.2) are increase/decrease based schemes in which receivers join/leave multicast groups in a coordinated manner. Both algorithms attempt to ensure fairness towards competing TCP traffic. The main operational logic is that subscription is decreased upon detection of packet loss and increased upon reception of sender-coordinated synchronization points.

Both RLC and FLID are based on the approximated throughput equation of TCP [71] [72]:

$$R = \frac{\sqrt{3/2}}{RTT\sqrt{p}}, \tag{5.6}$$

where RTT is round-trip time (seconds), p is packet loss fraction and R is throughput (bytes per second). The bandwidth partitioning and join/leave strategies of these two algorithms leads to a performance that approximates the throughput of equation 5.6 above. Well-behaved receivers in RLC and FLID are oblivious to this equation, and do not change their subscriptions based on it. Rather, they simply follow the increase/decrease strategies of the respective congestion control schemes. Since RTT is ill-defined in the multicast case, a fixed nominal RTT is used in the following.

A greedy receiver will not obey the sender-initiated synchronization points and will (by definition) violate the subscription strategy. However, this can be countered using the framework presented here. Substituting the TCP throughput equation (5.6) in equation 5.2, and optimizing the rate-distortion performance of the different subscriptions as explained earlier, there is no longer a gain in visual quality to be found from inflating subscription. This does not change the RLC/FLID protocols, since the extension here is merely a source/channel coding optimization that is done outside the scope of the congestion control algorithms. It is however mentioned the implementation presented here will only work with the static layering version of FLID.

5.6 Simulations and results

This section presents model-based simulations that highlight the intended performance and characteristics of the proposed framework. Specifically, the TCP throughput relation (equation 5.6) is used in place of equation 5.2. As discussed earlier, this is the relevant case for RLC and FLID. The same source and channel models as in the unicast case are used here. A best-effort network where all multicast groups are subject to equal probabilities and distributions of packet losses is assumed.

All simulations use the Gilbert channel model with $\rho = 0.1$. Packet size is 1250 bytes, an RTT of 100ms, and using the RDE algorithm of section 3.2.2 for optimization. In accordance with the bandwidth efficiency observed in figure 5.6, exponentially increasing subscription rates is employed. The lowest available subscription rate is 150 kbps, increasing by a factor of two for each subscription increase (i.e. 150, 300, 600, 1200 kbps and so on).

Figure 5.7 shows the obtained performance for three (a), four (b) and five (c) different subscription alternatives. The prefix parameter ξ is 0.2 for all three cases. Each plotted curve corresponds to a single subscription alternative, plotted as a function of packet loss fraction. The vertical dotted lines indicate the

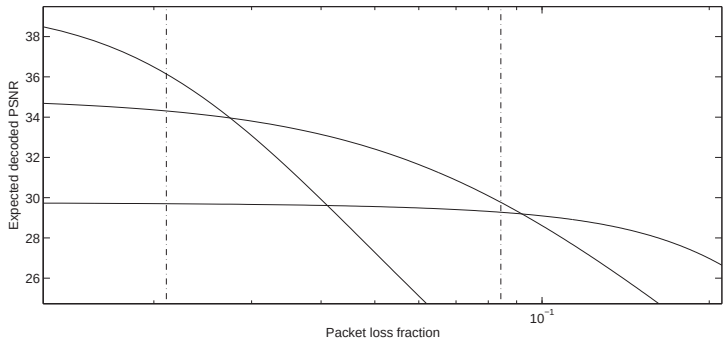
intended crossover points where a there should be a switch from one subscription to another. Thus, the PSNR performance curves of the different subscription alternatives should ideally intersect at these specific packet loss fractions. As the figure shows, the ability of the proposed algorithms to adjust the rate-distortion performance of the different video is good in most cases. As in the unicast case, the most important property to notice here is that, given a specific network state, the subscription alternative yielding the lowest distortion (highest PSNR) has a rate that is fair in a congestion control context.

Figure 5.8 shows the performance for different values of ξ. As explained in section 5.4, ξ is the parameter controlling the fraction of transmission rate used as *prefix* in figure 5.5. The figure shows the case of ξ equal to 0.15 (a), 0.25 (b) and 0.35 (c). As is clear from the figure, the ability of the algorithms to align the intersections of PSNR performance curves with the intended crossover points (vertical dashed lines) is poor for the lowest value of ξ. This aligning ability increases for larger values of ξ, although the improvement from $\xi = 0.25$ to $\xi = 0.35$ is marginal. The reason for the reduced ability of aligning the intersections for low values of ξ is that this gives few packets in the leaf groups. This gives a low granularity (few degrees of freedom) available for allocating channel codes by the optimization algorithm. An improved performance can be gained by reducing the packet size while maintaining the same rate in the leaf groups, giving a better granularity for allocation of channel codes. However, this may not be desirable from due to the increased overhead incurred, and the fact that a higher number of packets can increase congestion at routers. Furthermore, dealing with different packet sizes is problematic with respect to equation 5.6.

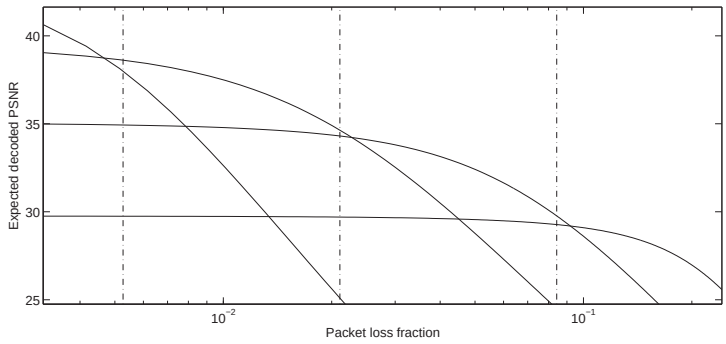
5.6.1 Discussion

Results indicate that the framework is able to provide a strong incentive for receivers to adjust subscriptions in keeping with fair rate sharing strategies. The accuracy in aligning the intended crossover points (vertical dashed lines in the result plots) between different subscription alternatives is seen to be dependent on the ξ parameter, as would be expected. Hence, there is tradeoff between accuracy of matching performance curve intersections with these crossover points and the resulting bandwidth efficiency of the scheme.

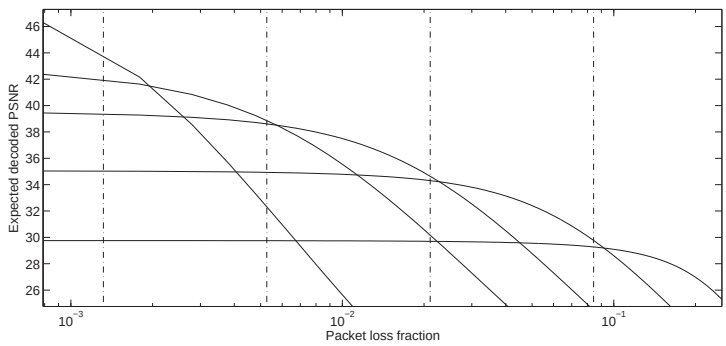
As is evident from the packet loss fraction axis in the plots, realistic rates for the subscription alternatives (with the considered RTT and packet size parameters) correspond to a large range of packet loss fractions. That is, the range of packet loss fractions corresponding to the chosen possible subscription rates (through the inverse of equation 5.2) will be relatively large. However, the nonlinearity of the TCP throughput equation yields very closely spaced crossover points for the highest rates. Thus, it is debatable whether the proposed system will exhibit the intended separation of resulting visual quality for the subscription alternatives with high rates. The reason for this is that, in typical networks,



(a)

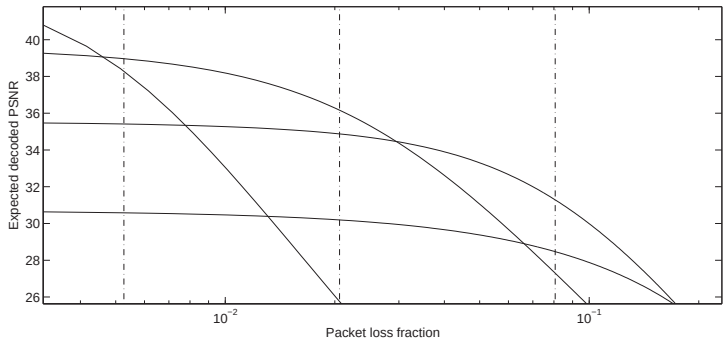


(b)

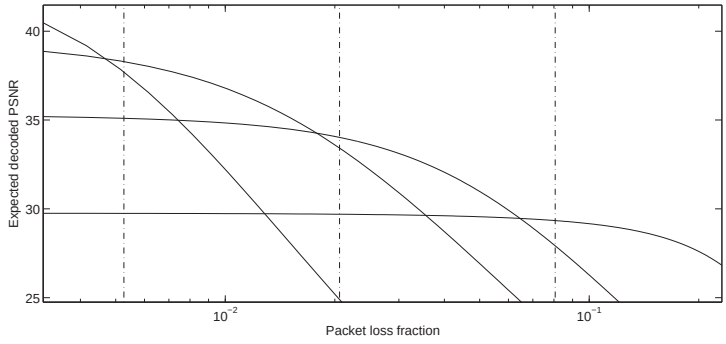


(c)

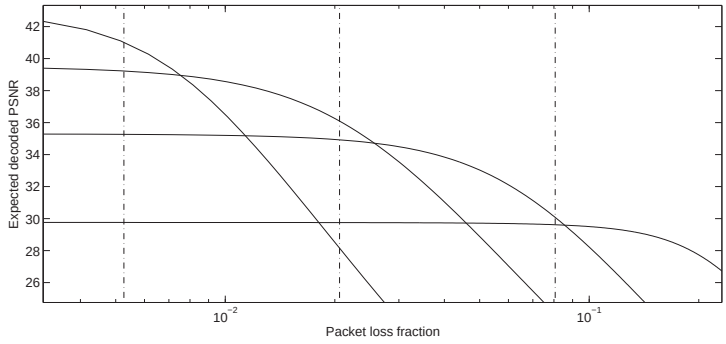
Figure 5.7: PSNR as a function of packet loss fraction for three (a), four (b) and five (c) subscription alternatives ($\xi = 0.2$). Each plotted line corresponds to a specific subscription alternative, and the vertical dashed lines show the intended "crossover" points between two subscription alternatives, as found via the TCP throughput equation.



(a)



(b)



(c)

Figure 5.8: PSNR as a function of packet loss fraction for $\xi = 0.15$ (a), $\xi = 0.25$ (b) and $\xi = 0.35$ (c). All plots are for the case of four subscription alternatives. Each plotted line corresponds to a specific subscription alternative, and the vertical dashed lines show the intended "crossover" points between two subscription alternatives.

the packet loss fraction may be seen to have an inherent variability larger than the fraction which should separate the subscription alternatives with the highest rates.

5.7 Conclusions and future work

A joint source-channel coding based framework for congestion control in multicast video communication has been presented. Similar to the unicast case, the framework provides an incentive in terms of visual quality for receivers to adjust their subscribed rates in accordance with fair bandwidth sharing strategies. Simulation results show that the intended performance is attained with good accuracy in matching congestion state to visual quality.

The proposed framework can use the classical cumulative layering approach. However, in order to provide a stronger incentive for receivers to reduce their subscription upon congestion, a new semi-cumulative layering scheme is proposed. Specifically, a certain data dependency across multicast groups is introduced in order to yield the intended performance. The new layering strategy incurs an (adjustable) reduction in bandwidth efficiency compared to the classical case, as well as a somewhat higher use of multicast addresses. Comments are given on how this bandwidth efficiency can be kept at a minimum.

Different from earlier proposed approaches, there is no need for any feedback from receivers to the sources. This novel feature is highly desirable in multicast delivery, as it avoids the potential *feedback implosion* problem and thus allows for the multicast sessions to scale to large audiences and large networks.

Recommendations for future work include:

- *Trials in real networks.* In order to validate the applicability and efficiency of the proposed approach, trials in real networks are needed. Of particular interest is the temporal extent over which the incentives for rate reduction in the case of congestion are in effect (as discussed in section 5.4).

- *Integration with dynamic layering.* The simulations shown here only consider *static* layering. For integration with rate control approaches that use *dynamic* layering (e.g. FLID-DL [72] and WEBRC [73], see section 2.3.2), further developlment is needed. Of perhaps particular interesting is to considering how the framework can be used together with schemes based on rateless codes (e.g. WEBRC).

- *Payload format.* As in the unicast case, use of the IETF RTP payload format [103] for MD-UXP data should be investigated.

Part C

Multisource video streaming in mobile ad hoc networks

The main ideas and results of this part of the thesis have been published as follows:

Thomas Schierl, Stian Johansen, Cornelius Hellge, Thomas Stockhammer and Thomas Wiegand — "Distributed rate-distortion optimization for rateless coded scalable video in mobile ad hoc networks". In *Proceedings of IEEE International Conference on Image Processing (ICIP-07)*, San Antonio, Texas, USA, September 2007.

Thomas Schierl, Stian Johansen, Andrew Perkis and Thomas Wiegand — "Rateless Scalable Video Coding for Overlay Multisource Streaming in MANETs". Submitted for publication in *Journal of Visual Communication and Image Representation, special issue on Resource-Aware Adaptive Video Streaming*, Elsevier B.V., July 2008.

Chapter 6

Multisource video streaming in mobile ad hoc networks

Recent advances in forward error correction and scalable video coding enable new approaches for robust, distributed streaming in Mobile Ad Hoc Networks (MANETs). This chapter presents an approach for distribution of real time video by uncoordinated peer-to-peer relay and/or source nodes in an overlay network on top of a MANET. The main novelty in the approach proposed here is that it allows for *distributed, rate-distortion optimized transmission-rate allocation* for competing scalable video streams at relay nodes in the overlay network. The approach has the desirable feature of path/source diversity that can be used for enhancing reliability in connectivity to serving nodes and/or attaining a higher throughput. The distributed approach reduces signaling overhead as well as avoiding scalability issues that come with centralized processing in MANETs. Results show a significant performance gain over both single-server systems and previously proposed multi-source systems.

6.1 Introduction

Recently, MANETs [74] (see section 2.4) based on the ad hoc mode of IEEE 802.11 WLAN [106] or the emerging IEEE 802.16j WiMAX[1] Mobile Multihop Relay [107] and IEEE 802.11s [108] standards have gained interest for delivery of multimedia content and other mobile services. Similar to 'push' services in 3G mobile networks, new services can be introduced based on ad hoc groups built on top of MANETs. MANETs are attractive due to low infrastructure costs, especially in areas with high user density. The coverage area for mobile services can generally be extended through cooperation with neighboring nodes.

[1] Worldwide Interoperability for Microwave Access

In MANETs, user terminals in a mobile network are conceptually not assumed to be receivers only, but can also be used as routing nodes in order to build a dynamic network infrastructure.

User nodes building an on-demand MANET are by definition assumed to be mobile, which results in highly dynamic characteristics for this type of network. Thus, a topology built upon a MANET cannot be truly robust against network separation, route/path losses and packet losses. Therefore, clients typically experience connection losses to serving nodes [109].

Multimedia delivery services in MANETs can be implemented using non real-time downloads or real-time streaming. Download delivery in general does not relate to the usual timing constraints for media data. By using appropriate end-to-end protocols (e.g. [110] [111]), one could more easily deal with connectivity loss and longer outages in MANETs, in order to provide full reliability. For real-time delivery, on the other hand, timely delivery is crucial. In this case, where the associated delay constraints [109] are of prime importance, reliability is much harder to achieve. Furthermore, the available throughput in MANETs is typically orders of magnitude lower than many other wireless (and certainly wired) networks, leading to increased congestion and contention. When simply using common point-to-point transmission techniques such as link layer forward error correction or retransmission protocols, sufficiently good service quality in MANETs is often not possible. Hence, solutions for satisfying the different connectivity requirements of real-time streaming in MANETs are needed.

The solution to this problem presented here is based on enhancing source connectivity by using source node diversity (i.e. streaming from multiple sources concurrently) combined with the use of a family of "rateless" forward error correction codes [12] [61]. The proposed approach exploits the benefits of cooperative interaction between peers in an overlay network on top of a MANET for maximizing video quality, adapting to varying network conditions and enhancing connectivity.

For improving application layer QoS, scalable video coding and application layer forward error correction is employed. In general, a scalable video stream allows for flexibility in rate allocation and adaptation at peers in the overlay network, as peer nodes may decide to forward or not to forward a network stream in order to adapt the transmission rate. By using an efficient and flexible FEC code, the Raptor code [61], in combination with scalable video coding, reception of real time video data from uncoordinated peers is realized.

The basic approach for distribution of media streams has been shown in [109], and is further extended here by using a separate FEC encoding process for each video layer, similar to the proposal in [112]. As will be shown, this allows for flexible adaptation of transmission rates and the ability to perform rate-distortion optimization in a distributed manner. Rate-distortion optimization in this context is somewhat different from that of parts A and B of this thesis. Here, it involves rate allocation for the different scalable video streams that are

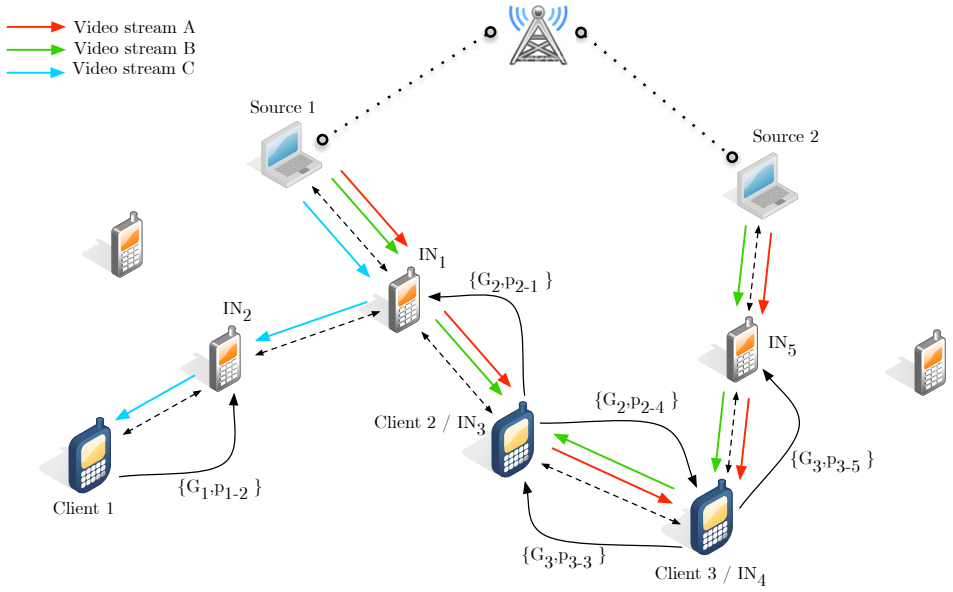


Figure 6.1: Overlay networks for MANET multiple source media distribution based on RSVC.

competing in the network [113]. The rate-distortion optimization approach takes into account local competing traffic, characteristics of video streams as well as connectivity information for clients. The fact that optimization is done in a distributed manner is a key requirement for operation in MANETs because of their dynamic nature.

In the following section, the general system model and its components are presented. Section 6.3 describes the rate-distortion optimization framework and its implementation. Section 6.4 describes the simulation setup, gives a selection of simulation results and provides a discussion of relevant issues with the presented approach. Section 6.5 concludes the chapter and provides suggestions for extensions of the proposed approach.

6.2 Media transport in MANETs

Figure 6.1 shows the basic view of an overlay network in a MANET as considered here. In the example scenario shown in the figure, three clients are receiving three potentially different video streams from two source nodes. It is assumed that the sources are fed by additional reliable access networks, here shown as a wireless downlink. Reliable access to video for the source nodes could in practice

be through any access network providing sufficient bandwidth. Even though a wireless connection is shown in the figure (perhaps most natural for MANETS), the sources could also be connected through a wired medium. Interesting wireless access methods could be digital terrestrial television (DVB-T[2] or the more ubiquitous IEEE 802.11 family of standards. For practical reasons, it is beneficial that the compressed video received by the sources is already scalably encoded. If necessary, transcoding from a non-scalable to a scalable representation can be done at the source nodes. The encoding from compressed video layers to streams of Raptor encoding symbols is done at the source nodes, using procedures described in the following. In the following, the proposed system will be referred to as RSVC, *Rateless Scalable Video Coding*.

As is the case in MANETs, the video streams for the three client nodes in the figure are relayed by intermediate nodes (INs), and it is also illustrated that client nodes themselves act as INs if necessary. Specifically, clients 2 and 3 in the figure are also used as INs.

It is mentioned that the use of multiple sources has similarities to the so-called "swarm" approach in peer-to-peer (P2P) networks. The swarm approach was initially introduced in [114] for large-scale P2P networks. In P2P networks, mesh-based approaches aim to construct an overlay network whose connections are maintained through "gossip" messages. In this case, peers are self-organized into a mesh and independently request portions of the video from neighbors, with no particular emphasis on the structure of the distribution path [115] [116].

6.2.1 Rateless codes

The Raptor code [61] (see section 2.2.2) is an efficient erasure correction code mainly used in environments with packet losses. The rateless/fountain property of the Raptor code implies that a virtually infinite amount of independent encoding (output) symbols (ESs) can be generated from a limited number of source (input) symbols (SSs). Transmitting these ESs intelligently over different paths using different sources can significantly enhance the reliability of streaming sessions in MANETs. For the multiple source case, a randomization mechanism has been proposed in [109] for making the different Raptor encodings at different sources linearly independent without the need for coordination among the sources. Because of this property, a Raptor decoder at a receiver does not need to be aware from which source an encoding symbol originates. Rather, the receiver only needs to concern itself with receiving a sufficient amount of encoding symbols in order to allow decoding.

[2]Digital Video Broadcasting - Terrestrial

6.2.2 Rateless scalable video coding

In [109], the generated FEC symbols of the different video layers are distributed into network packets based on Priority Encoding Transmission (PET) [10], in this case analogous to MD-FEC[3] (forward error correction based multiple description coding) [117] [118]. In [109], MD-FEC is implemented such that reception and decoding of a single MD-FEC stream (i.e. client connected to a single source) allows for decoding the base layer. Reception of multiple MD-FEC streams (i.e. client connected to multiple sources) allows for decoding the corresponding number of layers. This has the advantage of providing resilience toward route loss and video playback interruptions when connected to multiple sources, but is on the other hand increasingly wasteful of bandwidth as the number of sources increase.

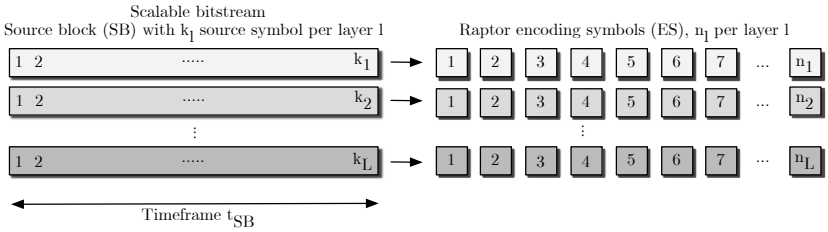
In the work presented here, the RSVC process is extended by transporting the different RSVC streams on different network transport streams, an approach similar to that of [112]. The rigid pre-defined structure of the MD-FEC streams is loosened, allowing the individual clients to subscribe arbitrary fractions of the RSVC streams from the different sources. Thus, the clients are able to optimize their subscriptions from sources based on connectivity, route reliability and experienced loss characteristics along the different paths.

Figure 6.2 shows the RSVC network stream encoding, transport and aggregation. A source block (SB) of source symbols corresponding to one time-frame of the scalable video data with duration t_{SB} is encoded with different Raptor encodings per video layer l. Using the Raptor code for encoding the k_l source symbols (SSs), this theoretically allows for producing an unlimited number n_l of encoding symbols (ESs) per source block per layer l. Assume that for a source block of length t_{SB}, a receiver receives $\tilde{m}_l^s$ encoding symbols from each source s for substream l, corresponding to video layer l. The efficiency of the Raptor code is such that if, on average, the sum of received symbols for layer l from S sources $\sum_S \tilde{m}_l$ is slightly greater than the number of SSs, k_l, video layer l within can be recovered [109]. Formally, the condition for being able to decode layer l is

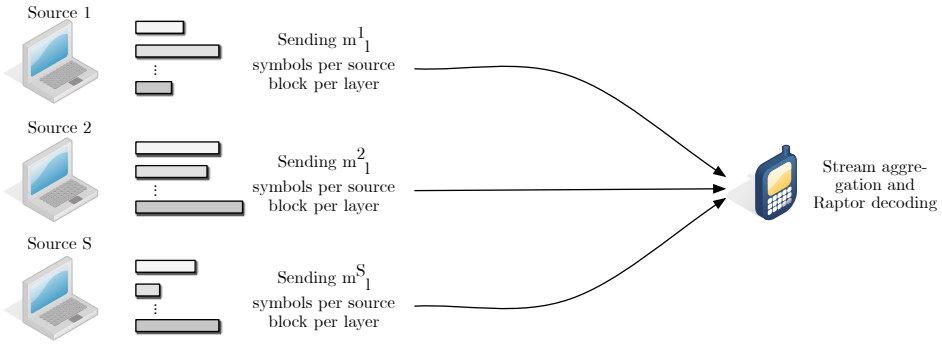
$$\sum_{s=1}^{S} \tilde{m}_l^s \;\geq\; (1+\epsilon)k_l, \tag{6.1}$$

where ϵ is the overhead of the Raptor encoding implementation. It is mentioned that the above implies that, when using a rateless channel code, a priori knowledge about channel loss characteristics is not needed. This is different from earlier distributed video streaming approaches, e.g. [119].

[3]It is noted that MD-FEC here is similar to the MD-UXP techniques discussed in part A of this thesis. However, MD-FEC in this context is not rate-distortion optimized for loss protection as in part A. Thus the term MD-FEC is used rather than MD-UXP in order to accentuate this difference.



(a) RSVC encoding from layered/scalable video to Raptor encoding symbols.



(b) Transport and aggregation of RSVC streams from multiple sources.

Figure 6.2: Rateless scalable video coding (RSVC)

Since a client potentially receives fractions of the video stream from multiple different source nodes, it is beneficial if the video streams received by the client are synchronized with source block (SB) accuracy. Looser synchronization is of course possible, but at the cost of larger receiver buffers at clients (and possibly increased delay). As is apparent from simulations presented later, synchronization on an SB level is achieved for the considered network sizes and topologies.

6.3 Distributed rate-distortion optimization for RSVC

This section presents the theoretical background and implementation details for the distributed optimization approach in RSVC. For ease of exposition, it is only referred to source-to-client connections in the remainder, since all connections of type source-to-relay, relay-to-relay and relay-to-client can be viewed as a source-to-client connection. The optimization procedure described in this section is valid for both source and relay nodes.

It is further assumed that the rate available for transmission on an overlay path is known. This may be achieved by techniques as proposed in [120], where time sharing and contention of the wireless channel at each MANET relay node is analyzed for estimating the available transmission rate. The estimation of available transmission rate along paths in a MANET is outside the scope of this work.

6.3.1 Rate-distortion optimized streaming of RSVC streams

The main aim of the optimization procedures described here is that the limited capacity between overlay nodes in MANETs is shared in such a way that the sum of video qualities experienced at receivers is maximized. The roles of client nodes and source/relay nodes are different in the system, and are explained separately in the following for clarity. The reader is referred to figure 6.1 in the following for a visualization of the message exchanges between source/relay nodes and client nodes.

Client node operation

When starting a video streaming session, the client node attempts to contact source nodes. In the simulations given in section 6.4 the source nodes are assumed to be a priori known to client nodes, but this can also be done by broadcasted requests (flooding) in the network. The reachable source/relay nodes will reply to the client with an acknowledgment message indicating their availability and the rate that has been allocated to the client node (source/relay operation is described in detail below). Based on the rates allocated to the client from all reachable sources, the client invokes algorithm 7.

Here, the rates $r_l, l \in [1, L]$ needed to decode are naturally related to $k_l, l \in [1, L]$ by the size of encoding symbols. The algorithm must be invoked on the following occasions:

- Initialization of the streaming session
- Reception of a rate allocation message from any of the connected servers, indicating that the rate available has changed. This may happen if other clients join or leave streaming sessions with the sources.
- Loss of connection to either of the sources.

Since clients decide what video layer rates to request from the different clients, the proposed system can be seen as partly receiver-driven. However, as will be apparent in the following, the rate that may be requested from each source/relay node is limited by the respective sources/relays.

An important parameter in the above is the metric $M_i,\ i \in [1, S]$ which signifies the reliability of the connection to source $i \in [1, S]$. The approach taken

Algorithm 7 Client node rate request procedure

Require: Allocated transmission rates $R_i,\ i \in [1, S]$ from S available sources
Require: A metric $M_i,\ i \in [1, S]$ quantifying the reliability of connection to source i
Require: Rates r_l needed to decode video layer $l \in [1, L]$

```
Initialize list of subscriptions t for the video layers
Initialize list of source nodes n based on metric M, ordered by decreasing reliability
for k = 1 to S do
    while rate R^k_req requested from source n_k less than available rate R_k do
        Starting from lowest video layer not already sufficiently subscribed (t_l < r_l), request min{(r_l − t_l), (R_k − R^k_req)} from source k.
        Increase R^k_req by the subscribed rate.
        Update the table of subscribed layer rates, t.
    end while
end for
Send layer-specific subscription rates to sources.
```

in the simulations presented in section 6.4 is to use the *hop count*, the number of IP nodes from source to client, as the metric. This simple metric is meaningful in MANETs since the probability of a route loss (on average) increases with the number of IP relays involved in the communication. Other more complex metrics can be used, e.g. taking into account the mobility of involved relay nodes if such information is available.

In addition to the above, client nodes send status messages to all connected sources regularly to indicate connectivity in terms of experienced goodput G and packet loss p on the path to the source in question. These messages are indicated in figure 6.1, and are used by source/relay nodes for performing rate-distortion optimization.

Source/relay node operation

Consider the case where a source/relay node has N connected clients that are requesting video streams concurrently. The node then needs to divide the rate available for transmission between these connected clients. For this, algorithm 8 is invoked.

The distortion characteristics d_l^i quantify the distortion experienced when successfully decoding l video layers of the video stream communicated to client i. These discrete distortion points are obviously given by the video encoder, and need to be communicated to source/relay nodes as side information in the video streams. In the implementation simulated in section 6.4, these distortion points are included in the Raptor encoded video stream. This strategy incurs negligible

Algorithm 8 Rate allocation procedure at source/relay nodes

Require: The number of connected clients N and their associated goodputs G_i $i \in [1, N]$.
Require: Packet loss fractions p_i on paths to client $i \in [1, N]$.
Require: Rate-distortion characteristics d_l^i of all layers $l \in [1, L]$ of video communicated to client nodes $i \in [1, N]$;
1: Perform rate-distortion optimization based on equation 6.5 below.
2: Send rate allocation messages to all connected clients

overhead for a reasonably low number of video layers, since distortion points for each layer are averages over a source block (see figure 6.2), and have only to be communicated once per source block.

Algorithm 8 needs to be invoked whenever a client joins/leaves the streaming session, as well as whenever a message containing new goodput and path packet loss values for a client are received.

Rate-distortion optimization

Consider the situation where N client nodes are requesting video streams from a source node. Without loss of generality we assume that each client is receiving exactly one video stream. Further assume that the sending capacity at the source s in question is limited to $R_{s,avail}$, and that $R_{s\to n}^{path}$ denotes the available rate on the path to client n. Both values are assumed to be known to the source node (e.g. estimated using the method in [120]). As mentioned above, a video stream is characterized by a set of distortion points d_l, which represent some measure of the difference between encoded and original video (typically MSE), with r_l^{enc} being the corresponding encoding rate of the scalable video stream at layer $l \in \{1, \cdots, L\}$. A function D maps the total goodput g being received by a client to discrete distortion points d_l of the video stream received by the client in question.

$$D : g \to d \quad g \in \mathbb{R}^+ \quad d \in \{d_1, \cdots, d_L\} \tag{6.2}$$

Goodput is defined as

$$g_n = \sum_{s=1}^{S} (1 - p_{s\to n}) r_{s\to n} \tag{6.3}$$

for client n receiving data from S sources, each sending at rates $r_{s\to n}$, over paths characterized by packet loss fractions $p_{s\to n}$. Note that the above only holds for constant packet size, which is typically the case for the FEC-encoded packets used in this work. It is assumed in the following that the packet loss fraction $p_{s\to n}$ is known and independent of the transmit rate. Taking into account the overhead

factor ϵ incurred by the Raptor encoding, the following relations between goodput g and distortion points d_l holds:

$$D = \begin{cases} d_0 & \text{when } g < (1+\epsilon) r_1^{enc} \\ d_l & \text{when } g \geq (1+\epsilon) r_l^{enc} \text{ and } g < (1+\epsilon) r_{l+1}^{enc},\ l \in [1, L-1] \\ d_L & \text{when } g \geq (1+\epsilon) r_L^{enc} \end{cases} \tag{6.4}$$

Using the above, the source attempts to minimize the average distortion experienced at all connected clients as follows:

$$\min_{\{\Delta r_{s\to 1}, \Delta r_{s\to 2}, \cdots, \Delta r_{s\to N}\}} \left(\sum_{n=1}^{N} D_n(g_{n,opt}(\Delta r_{s\to n})) \right) \tag{6.5}$$

As is evident from equation 6.5, the parameters subject to optimization are the set of transmission rate changes $\{\Delta r_{s\to 1}, \Delta r_{s\to 2}, \cdots, \Delta r_{s\to N}\}$ for the N clients. The optimal goodput $g_{n,opt}$ for client n, as stated in equation 6.6 below, is calculated with $r_{s\to n}$ being the rate the source s is sending to client n, and $\Delta r_{s\to n}$ being the change in allocated rate.

$$g_{n,opt} = \overbrace{g_n - \underbrace{(1-\gamma_{s\to n}) r_{s\to n}}_{\text{Old goodput from this source}}}^{\text{Goodput from other sources}} + \underbrace{(1-\gamma_{s\to n})(r_{s\to n} + \Delta r_{s\to n})}_{\text{New goodput from this source}} \tag{6.6}$$

which simplifies to

$$g_{n,opt}(\Delta r_{s\to n}) = g_n - (1-\gamma_{s\to n}) \Delta r_{s\to n} \tag{6.7}$$

Optimization is done under constraints (6.8) through (6.10).

$$\sum_{n=1}^{N} (r_{s\to n} + \Delta r_{s\to n}) \leq R_{s,avail} \tag{6.8}$$

$$r_{s\to n} \leq R_{s\to n}^{path} \tag{6.9}$$

$$(1-\gamma_{s\to n}) r_{s\to n} \leq \Delta r_{s\to n} \leq R_{s\to n}^{path} - r_{s\to n} \tag{6.10}$$

Here, (6.8) constrains the rate increase at source node s, (6.9) restricts the rate on the path to the receiver, and condition (6.10) gives the upper and lower bounds on the rate change $\Delta r_{s\to n}$.

When relay nodes have carried out the RSVC rate-distortion optimization described above, information about the allocated rates $r_{s\to n},\ n \in [1, N]$ is propagated to the connected clients. Based on these messages, clients decide which rates for each media layer should be requested from each overlay node (as in algorithm 7). In other words, the client is partitioning its total allocated rate

to subscriptions for the video layers at the available overlay nodes in order to minimize distortion and/or maximize reliability.

This optimization procedure fulfills two important aspects of MANET communication: Cooperation and distributed processing. Each participating node carries out its own optimization and propagates the decisions to the other nodes who, in turn, use it for their local optimization.

Stability

The heuristic algorithm described above depends simultaneously on the dynamics of the network and the dynamics of the video streams. A potential problem can be encountered if the rate-distortion optimizations at source/relay nodes are done in approximate synchronicity. This may lead to oscillations over time in the allocated rates to client nodes. In the implementation simulated here, this is avoided by scheduling the optimizations at servers with a random delay after a change in the network is discovered or reported. In this way, changes in allocation from a source node are likely to be reflected in the goodput reports from the client in question to the the other connected source nodes before they invoke optimization. In this way, optimizations based on identical information is avoided, thus greatly reducing the likelihood of oscillations. Extensive simulations show that stable operation is achieved.

6.4 Simulations and results

This section presents a set of selected simulation results that highlight the performance of the system. The proposed system is compared to the MD-FEC based approach of [109] and the case of using single-server streaming. The system has been integrated into the ns-2 network simulation environment [121] presented in [109].

6.4.1 Source material, encoding and R/D characterization

The three different ITU-T video sequences (repeated forward and backward) City, Crew and News in QCIF resolution (Quarter Common Intermediate Format, 176 x 144 pixels) were used in the simulations. The sequences were encoded at 15 frames per second, repeated forwards and backwards yielding a total length of approx. 100sec. All sequences are encoded using the Scalable Video Coding (SVC) extensions of H.264/AVC [4] [8]. The JSVM[4] 8.8 reference software [122] was used for encoding, using an H.264/AVC base layer and four SVC fidelity enhancement layers (ELs) with medium-grain fidelity scalability (MGS) [8]. A group-of-picture (GOP) size of 16 was used, having one IDR (Independent

[4] Joint Scalable Video Model

	City		Crew		News	
	Rate	PSNR	Rate	PSNR	Rate	PSNR
Base layer	53.4	34.5	58.8	29.2	45.0	36.1
Enh. layer 1	81.8	37.2	77.0	31.0	69.2	38.7
Enh. layer 2	88.9	37.4	87.4	31.4	78.7	39.4
Enh. layer 3	142.3	39.9	125.8	33.5	128.5	42.4
Enh. layer 4	169.4	40.9	150.4	34.7	159.5	44.3

Table 6.1: PSNR and rate values for base layers and enhancement layers for the three transmitted SVC encoded video sequences

Decoder Refresh) frame in each GOP for random access. All streams are encoded at a rate of about 160kbit/sec (cumulative rate of all layers). The rate points (layer rates) are achieved by removing NAL units of the enhancement layer from the bi-stream starting with the lowest temporal priority. The resulting PSNR values are shown in table 6.1.

One source block (see figure 6.2) was generated every two GOPs, i.e. the minimum adaptation interval is about 2.13 sec. Raptor performance is evaluated by applying the simulation approach introduced in [110]. Each video layer has been encoded within an emulated, independent non-systematic Raptor encoding process. Thus, the resulting streams are decodable independently. For Raptor encoding, the 3GPP[5]-recommended preconditions are used [123]. A pre-buffering for network jitter compensation of 5 seconds is assumed.

6.4.2 Simulation details

For simulations, 20 different (random) MANET scenarios were used. Each scenario has 30 mobile nodes moving on random waypoint patterns at a maximum speed of 3m/s within an area of 650x650m. In each scenario the number of available sources was kept constant, with each source having a fixed (maximum) sending rate. Client nodes were selected randomly, varying the number of clients from 2 to 5. For each different number of clients in the system, a simulation lasting approx. 100s was done for each of the 20 scenarios. Results are found as averages over all scenarios, meaning that each data point shown in the result plots corresponds to an average over a simulation time of approx. one half hour.

Each client node selects a video stream from the set of available sequences (see table 6.1) in a round-robin fashion. Specifically, the first client requests the 'Crew' sequence, the second the 'News' sequence, the third 'City' sequence. This is then repeated from the 'Crew' sequence if more than three clients are present. The metric described in section 6.3 is used as basis of the rate request algorithm.

[5]3rd Generation Partnership Project

For comparison, two other transport methods are simulated as well as the proposed method. Specifically, the MD-FEC approach of [109] and a state-of-the-art single server approach with rate adaptation were simulated. In the single-server case, each client is restricted to be connected to one source at a time, but chooses the most reliable (based on the described metric) of all available servers. Furthermore, if a source in this case has more than one connected client, the available rate is divided equally among clients. In the simulations, both the MD-FEC and the single-server case were simulated using the same scalably encoded video sources as the proposed approach.

Throughput limitations on the paths through the overlay are emulated by enforcing transmission rate limitations at the serving overlay nodes. It is emphasized that, as noted earlier, the available bandwidth on an overlay path could be dynamically estimated as proposed in [120].

It is also mentioned that the OLSR (Optimized Link State Routing, see section 2.4.1) protocol [81] was used in the performed simulations.

6.4.3 Results

Figures 6.3, 6.4 and 6.5 show average received video quality over all clients in the overlay in terms of PSNR for the three simulated schemes. In figure 6.3, there are two servers in the topology, each providing a maximum total rate of 160kbps. In figure 6.4 there is also two servers in the topology, but here each server is able to provide a total rate of 240 kbps. Figure 6.5 shows the case where there are three servers available in the topology, each providing a rate of 160kbps. All of these three figures show how the sum total PSNR experienced in the system depends on the number of client nodes present. RSVC denotes the distributed RD-optimized method proposed here, MD-FEC refers to the method of [109] and SINGLE refers to the system constrained to single-server video streaming. It is mentioned that, in order to make the comparison fair, both SINGLE and MD-FEC approaches use the same scalably encoded video streams as the RSVC approach, but without rate-distortion optimization at source/intermediate nodes.

For comparison, the result plots also include the performance that would be attained had all clients been connected to all servers throughout the simulations (the ideal case). This is labeled EQUAL in the plots, indicating that the total available rate is equally partitioned among clients. It is emphasized that the EQUAL case is an idealized scenario with full connectivity at all times It is thus *calculated* — not simulated. For the RSVC, MD-FEC and SINGLE cases, simulation results will reflect connectivity restrictions induced by mobility and topology.

Results show that the RSVC approach performs consistently better than the the MD-FEC and SINGLE approaches. When the number of clients and the diversity of video streams increases, there are more degrees of freedom for doing the RD-optimization. Therefore, the performance gain of RSVC over the other

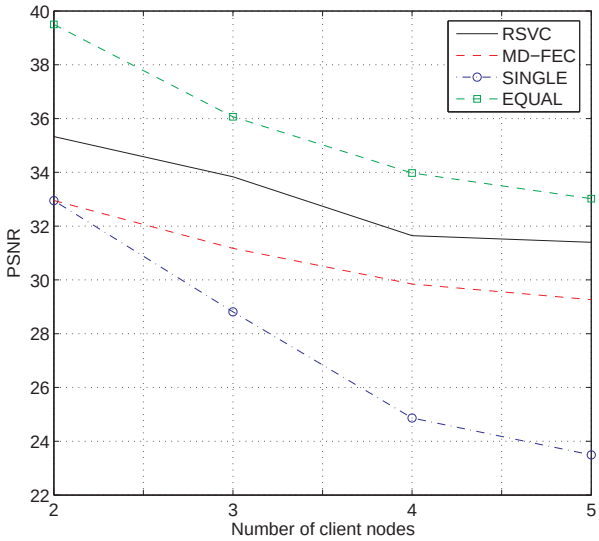


Figure 6.3: Average PSNR as a function of the number of clients. There are two source nodes in the topology, each providing a maximum rate of 160kbps.

methods generally increases with the number of clients. Due to the connectivity-preserving property of MD-FEC, it generally performs better than the SINGLE approach, but has a lower performance than the SINGLE approach when number of clients is low — due to the MD-FEC rate overhead. The RSVC approach gives a significant average performance gain over the other two systems, since the connectivity of clients and rate-distortion information about the video streams is taken into account.

As would be expected, all approaches exhibit decreasing PSNR as the number of clients increase. This is because of the fact that the rate available at sources (and along overlay paths) is limited, yielding a decreasing attainable throughput for each client as the number of clients increases. It is also seen that the performance of the idealized EQUAL case is not attained, since this case assumes full connectivity to all sources for all client nodes at all times. It is however seen that the RSVC case has a performance relatively close to the EQUAL case, especially for figure 6.3. The seemingly peculiar behaviour non-monotonous decrease of the EQUAL curve in figures 6.4 and 6.5 is due to the varying PSNR values for the different video streams. Specifically, the fifth receiver in the system is requesting the 'News' video sequence that exhibits the best PSNR performance (see table 6.1).

Figure 6.6 shows how PSNR develops over time for an example scenario with three servers and three clients present in the topology. As the figure

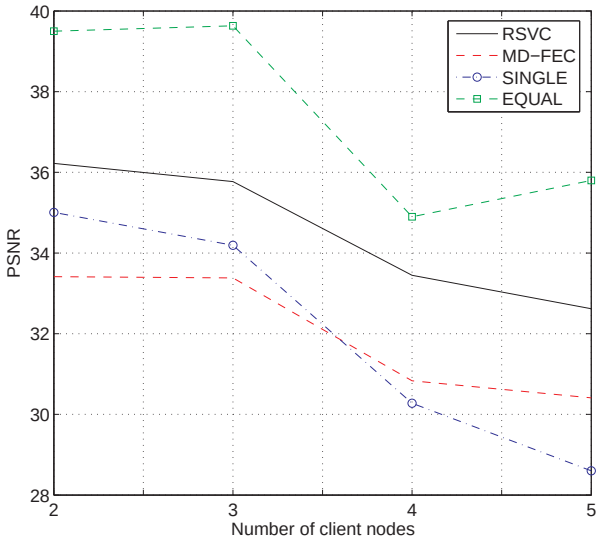


Figure 6.4: Average PSNR as a function of the number of clients. There are two source nodes in the topology, each providing a maximum rate of 240 kbps.

shows, the rate-distortion optimizations done at serving nodes implies that the PSNR experienced at the individual clients will fluctuate over time. Of course, these fluctuations come in addition to fluctuation resulting from the the video encoding itself. Also, the plot shows an example scenario where one of the clients experiences an outage due to route loss near the end of the simulation. It is noted that the PSNR indicated where client 1 experiences an outage is the average freeze-frame PSNR for a source block of the encoded video stream.

6.5 Conclusions and research directions

An approach for robust real-time video transmission in MANETs is presented. The approach uses a rateless forward error correction code in combination with scalable video coding for distribution of layered video to different sources in an overlay network on top of a MANET. In particular, a distributed mechanism for rate allocation at relay nodes is presented. The rate allocation and by that the adaptation of the scalable video stream is done in a rate distortion optimized manner. That is, information about the rate distortion characteristics of the layered video as well as the connectivity of clients is taken into account in order to minimize overall distortion experienced at connected clients.

Results indicate that the proposed system has significant advantages over single-server streaming approaches in general, as well as earlier proposed

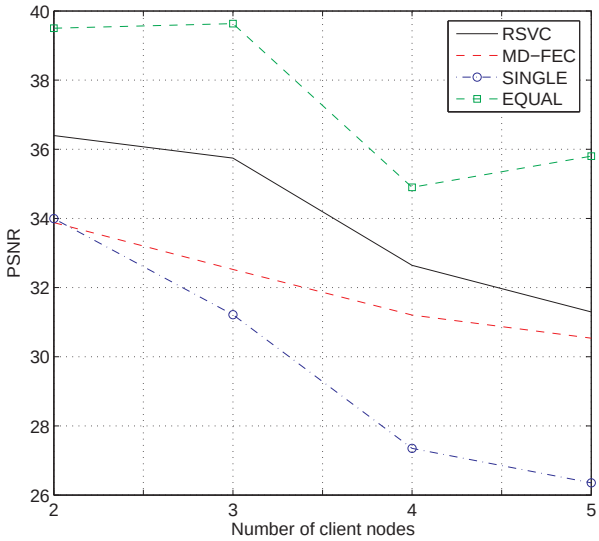


Figure 6.5: Average PSNR as a function of the number of clients. Here, there are three source nodes available in the system, each providing a maximum of 160 kbps.

multisource streaming solutions. The performance gain over the single-server streaming case is seen to increase as the number of clients increases, while the gain over the MD-FEC approach is less dependent on the number of clients in the system.

Ongoing and future work includes the integration of the proposed system with contention-aware routing. This has the potential for significant gains in MANETs, due to overlapping interference ranges of communicating nodes and its influence on CSMA/CA protocols. Thus, in order to achieve a higher throughput, traffic should either be concentrated along a small number of paths or along paths that have minimum overlap in terms of interference range.

Online estimation of available transmission rates along paths is also crucial for exploiting the available resources as well as avoiding contention issues, and needs to be integrated in practical systems.

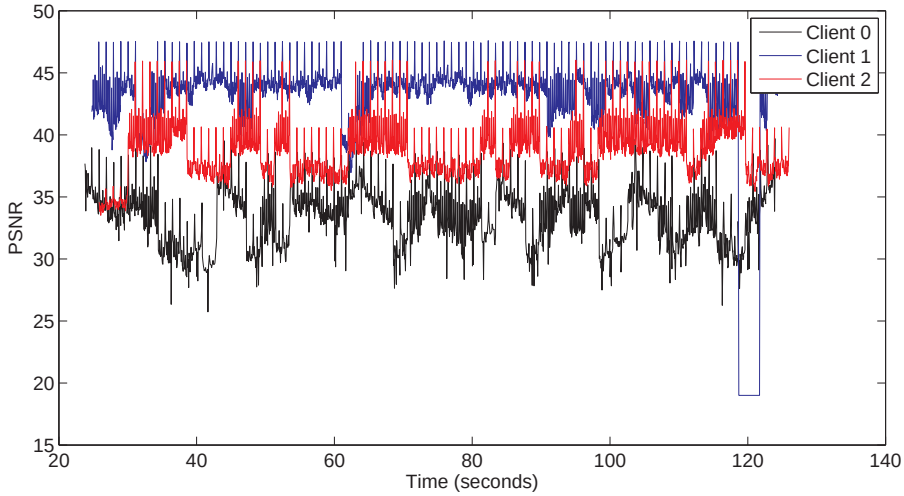


Figure 6.6: PSNR over time for clients. In this example scenario, there are three servers and three clients present in the topology.

Chapter 7

Conclusions

This thesis has dealt with video communication over resource constrained IP networks. Of particular importance has been the case of bandwidth-constrained networks in which a multitude of users compete for bandwidth. This scarcity of bandwidth may lead to congestion problems, inducing unwanted delays and losses of data. Ambient packet losses may also be experienced due to physical properties of the transmission channel.

When communicating video in these types of networks, it is of great importance to consider the effect of losses and delays on the visual quality of the video streams. **Part A** of the thesis investigated means of protecting the video data in an optimized manner. By *optimized* it is in this context considered how to optimally allocate the available transmission rate to the transport of encoded video data and its associated error correcting codes — in order to minimize expected distortion in the received video. The emphasis of the presented work has been on communicating over multiple parallel channels that may be logically or physically different. Four different algorithms were investigated and compared in terms of performance (i.e. expected distortion) and computational complexity. The results give insights into the various parameters that influence performance in the multichannel communication case, as well as a basis for selecting the optimization algorithm best suited for specific applications or network configurations. Of particular interest is the proposed rate-distortion optimized equal loss protection algorithm. This algorithm exhibits performance close to that of the more complex considered algorithms despite its relatively low computational complexity.

Delay constrained video communication services typically use the UDP protocol, an unreliable protocol that, unlike TCP, does not in itself implement congestion control. Large-scale deployment of video services requires implementing congestion control in order to not yield significantly unfair bandwidth sharing with TCP-based services. Nevertheless, there exists an incentive for

receivers in a shared network to either *not* implement congestion control or to *intentionally* attempt to obtain an unfair bandwidth share. This is especially relevant with the emergence of more error resilient video coding schemes that can tolerate significant packet losses (resulting from congestion) without experiencing a breakdown of video quality. To address this potential problem of receiver misbehavior, **Part B** of the thesis considered video quality based incentives for inciting proper congestion control. Through the presented framework, source coding, channel coding and congestion control are considered *jointly* in order to accurately match the expected video distortion to the congestion state of the network. The goal of this work is to reverse the current incentive for receiver misbehavior. In the proposed framework and its implementation, a misbehaving receiver will experience an *increase* in video distortion rather than a decrease as the result of misbehavior. Consequentially, there is no longer a disparity between proper rate adaptation (congestion control) and (selfish) maximization of video quality by receivers. The framework is developed for both the cases of unicast and multicast communication, using the optimization algorithms of Part A in its implementation. For the multicast case, a novel layering scheme is also presented; this can be used instead of the classical cumulative layering scheme for providing a stronger incentive for proper rate adaptation by receivers. Results show that the proposed framework and its implementation is able to remove the existing incentive for receiver misbehavior and replace this with an incentive for proper rate adaptation in the event of congestion. The accuracy in matching congestion state to distortion performance is seen to be good (although depending on channel models), and having the fortunate side-effect that distortion is minimized for well-behaved users.

Another dimension related to the above is how characteristics of video streams and connectivity of heterogeneous users can be considered jointly in order to provide performance gains in terms of video quality. **Part C** of the thesis concerned itself with this type of context-aware optimization. Specifically, the case of multiple users concurrently streaming different video streams in highly bandwidth-constrained mobile wireless ad-hoc networks is considered. The proposed approach uses a combination of scalable video coding, server diversity and multipath video streaming. At the core of the proposed approach is a distributed rate-distortion optimization scheme, in which servers and relay nodes attempt to *optimally partition* the limited available transmission rate in order to yield *maximum overall video quality* in the network. By using rateless channel codes in combination with scalable video coding, the optimization can be done in a truly distributed fashion without the requirement for synchronicity between video sources. The features of distributed optimization and lack of synchronicity are of prime importance in mobile ad hoc networks, as this type of networks exhibits frequent route losses, network separation and highly topology dependent delays. The proposed approach allows for client nodes to make a trade off between maximization of video quality and resilience towards route

losses (and their potential effect of outages in video playback). Simulation results show a significant (several dB PSNR) performance gain over previous proposed multisource streaming approaches, as well as over single source approaches.

References

[1] "YouTube" — Google Inc., Mountain View, CA, USA. [Online]. Available: http://www.youtube.com

[2] Ellacoya Networks, Inc., "Ellacoya data shows web traffic overtakes peer-to-peer (p2p) as largest percentage of bandwidth on the network," Press release, June 2007.

[3] British Broadcasting Corporation (BBC), "iPlayer." [Online]. Available: http://www.bbc.co.uk/iplayer/

[4] ITU-T and ISO/IEC, "Information technology - coding of audio-visual objects - part 10: Advanced video coding [H.264]," ITU-T recommendation H.264(2008), Advanced video coding for generic audiovisual services. ISO/IEC 14496-10:2008 Information technology — Coding of audiovisual objects — Part 10: Advanced video coding (4th Edition), 2008.

[5] "Information technology – multimedia framework (MPEG-21) – Part 1: Vision, Technologies and Strategy," ISO/IEC 21000-1, 2004.

[6] "Information technology - generic coding of moving pictures and associated audio information: Video," ISO/IEC 13818-2, 2000.

[7] G. Cote, B. Erol, M. Gallant and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–866, November 1998.

[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep 2007.

[9] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, Jul 2003.

[10] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Symposium on Foundations of Computer Science*, pp. 604–612, Aug 1994.

[11] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Generalized multiple description coding through unequal loss protection," *IEEE International Conference on Image Processing*, vol. 1, pp. 411–415, Oct 1999.

[12] D. J. C. MacKay, "Fountain codes," *IEEE Proceedings - Communications*, vol. 152, no. 6, pp. 1062–1068, Dec 2005.

[13] "Transmission control protocol (TCP) - protocol specification," Darpa Internet Program, RFC 792, September 1981.

[14] "User datagram protocol (UDP)," IETF RFC 768, August 1980.

[15] "RTP: A transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.

[16] "TCP friendly rate control (TFRC): Protocol specification," IETF RFC 3448, January 2003.

[17] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson, "TCP congestion control with a misbehaving receiver," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 5, pp. 71–78, Oct 1999.

[18] M. Georg and S. Gorinsky, "Protecting TFRC from selfish receiver," *Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services*, pp. 57–61, Oct 2005.

[19] S. Gorinsky, S. Jain, H. Vin, and Y. Zhang, "Design of multicast protocols robust against inflated subscription," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 249–262, Apr 2006.

[20] I. T. Union, "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines," ITU-T recommendation T.81 (JPEG).

[21] G. Karlsson and M. Vetterli, "Sub-band coding of video for packet networks," *SPIE Optical Engineering*, no. 7, pp. 574–586, July 1988.

[22] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–337, Sep 1910.

[23] J. R. Ohm, *Multimedia Communication Technology - Representation, Transmission and Identification of Multimedia Signals*, ser. Signals and communication technology. Springer, 2004.

[24] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, Feb 1992.

[25] "Information technology - JPEG 2000 image coding system: Core coding system," ISO/IEC 15444-1, 2004.

[26] J.-R. Ohm, M. V. D. Schaar, and J. W. Woods, "Interframe wavelet coding - motion picture representation for universal scalability," *Signal Processing: Image Communication*, vol. 19, pp. 877–908, Oct 2004.

[27] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, Dec 2005.

[28] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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, Jul 2003.

[29] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, Jul 2003.

[30] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video-compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 587–597, Jul 2003.

[31] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–144, Aug 2006.

[32] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC fidelity range extensions: Tools, profiles, performance and application areas," *Proceedings of IEEE International conference on Image Processing*, vol. 1, pp. 593–596, Sep 2005.

[33] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, Jul 2003.

[34] "Information technology - generic coding of moving pictures and associated audio information: Systems," ISO/IEC 13818-1, 2000.

[35] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG-4 AVC extension," *IEEE International Conference on Image Processing*, pp. 161–164, Oct 2006.

[36] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec 1993.

[37] S.-H. Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Processing: Image communication*, vol. 16, no. 8, pp. 705–724, May 2001.

[38] P. Chen and J. W. Woods, "Bidirectional MC-EZBC with lifting implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 10, pp. 1183–1194, Oct 2004.

[39] S.-T. Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," *Proceedings of the MPEG-4 Workshop and Exhibition at ISCAS 2000, Geneva*, May 2000.

[40] A. Perkis, "On the importance of error resilience in visual communications over noisy channels," *Birkhauser Boston Journal on Circuits, systems and signal processing*, vol. 20, no. 3, pp. 415–446, Mar 2001.

[41] O. I. Hillestad, "Evaluating and enhancing the performance of IP-based streaming media services and applications," Ph.D. dissertation, Norwegian University of Science and Technology, 2007.

[42] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley and Sons, Inc., 1991.

[43] "An architecture for differentiated services," IETF RFC 2475, December 1998.

[44] "Information technology - coding of audio-visual objects - part 2: Visual," ISO/IEC 14496-2, 2004.

[45] S. Kumar and L. Xu, "RVLC decoding scheme for improved data recovery in MPEG-4 video coding standard," *Journal on Real-time Imaging*, no. 10, pp. 315–323, Oct 2004.

[46] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, Sep 2001.

[47] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," *Conference record of the thirty-third Asilomar conference on signals, systems and computers*, vol. 1, pp. 342–346, Jan 1999.

[48] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," *Proceedings of Data Compression Conference (DCC)*, pp. 13–22, Mar 1999.

[49] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 821–834, Mar 1993.

[50] C. Tian and S. S. Hemami, "Universal multiple description scalar quantization: analysis and design," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2089–2102, Sep 2004.

[51] Y. Wang, M. T. Orchard, and A. R. Reibman, "Optimal pairwise correlating transforms for multiple description coding," *Proceedings of IEEE International Conference on Image Processing*, pp. 679–683, Dec 1998.

[52] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Distributed video streaming using multiple description coding and unequal error protection," *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 849–861, Jul 2005.

[53] Y.-C. Lee, J. Kim, Y. Altunbasak, and R. M. Mersereau, "Layered coded vs. multiple description coded video over error-prone networks," *Signal Processing: Image Communication*, vol. 18, no. 5, pp. 337–356, May 2003.

[54] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in the H.264/AVC standard," *Journal on Visual Communcation and Image representation*, vol. 17, no. 3, pp. 425–450, Mar 2006.

[55] O. I. Hillestad, O. Jetlund, and A. Perkis, "RTP-based broadcast streaming of high definition H.264/AVC video: An error robustness evaluation," *Journal of Zheijang University (SCIENCE A)*, vol. 7, supplement 1, pp. 19–26, May 2006.

[56] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes (parts 1 and 2)*, ser. North-Holland Mathematical Library. North-Holland Publishing Company, 1977, vol. 16.

[57] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge University Press, 2003.

[58] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, Jun 1960.

[59] B. Sklar, "Introduction to reed-solomon codes," *Digital Communications: Fundamentals and applications (2nd edition)*, Apr 2001.

[60] M. Luby, "LT codes," *43rd Annual IEEE symposium om foundations of computer science*, pp. 271–282, Nov 2002.

[61] A. Shokrollahi, "Raptor codes," *IEEE Transactions on information theory*, vol. 52, no. 6, pp. 2551–2567, Jun 2006.

[62] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," *Proceedings of SIGCOMM*, pp. 43–56, Aug 2000.

[63] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *Proceedings of ACM SIGCOMM'98*, pp. 303–314, Aug 1998.

[64] K.-W. Lee, T.-E. Kim, and V. Bharghavan, "A comparison of two popular end-to-end congestion algorithms: The case of AIMD and AIPD," *In proceedings of IEEE Global Telecommunications Conference (GLOBECOM'01)*, vol. 3, pp. 1580–1584, Nov 2001.

[65] "Datagram congestion control protocol (DCCP)," IETF RFC 4340, March 2006.

[66] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion control without reliability," *Proceedings of ACM conference on applications, technologies, architectures and protocols for computer communications*, pp. 27–38, Sep 2006.

[67] "Profile for DCCP congestion control ID 2: TCP-like congestion control," IETF RFC 4341, March 2006.

[68] "Profile for DCCP congestion control ID 3: TCP-friendly rate control (TFRC)," IETF RFC 4342, March 2006.

[69] "Host extensions for IP multicsting," IETF RFC 1112, August 1989.

[70] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," *Proceedings of ACM SIGCOMM'96*, vol. 26, no. 4, pp. 117–130, Aug 1996.

[71] L. Vicisano, J. Crowcroft, and L. Rizzo, "TCP-like congestion control for layered multicast data transfer," *Proceedings of IEEE INFOCOM, Conference on Computer Communications*, pp. 996–1003, Mar 1998.

[72] J. W. Byers, G. Horn, M. Luby, M. Mitzenmacher, and W. Shaver, "FLID-DL: congestion control for layered multicast," *IEEE Journal on selected areas in communications*, vol. 20, no. 8, pp. 1558–1570, Oct 2002.

[73] M. Luby, V. K. Goyal, S. Skaria, and G. B. Horn, "Wave and equation based rate control using multicast round trip time," *Proceedings of ACM SIGCOMM*, pp. 191–204, Aug 2002.

[74] IETF, "Manet working group charter." [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[75] N. Abramson, "The Aloha syste - Another alternative for computer communicatons," *Proceedings of AFIPS Conference*, pp. 295–298, 1970.

[76] M. Frodigh, P. Johansson, and P. Larssonn, "Wireless ad hoc networking - the art of networking without a network," *Ericsson Review*, no. 4, pp. 248–263, Jan 2000.

[77] IEEE, "The 802.11 WLAN working group." [Online]. Available: http://www.ieee802.org/11/

[78] "The Bluetooth Special Interest Group." [Online]. Available: https://www.bluetooth.org/

[79] "Ad hoc on-demand distance vector (AODV) routing," IETF RFC 3561, July 2003.

[80] "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," IETF RFC 4728, February 2007.

[81] "Optimized link state routing (OLSR)," IETF RFC 3626, October 2003.

[82] IEEE Standards Assosciation, "Get IEEE 802." [Online]. Available: http://standards.ieee.org/getieee802/

[83] J. L. Sobrinho, R. de Haan, and J. M. Brazioo, "Why RTS-CTS is not your ideal wireless LAN multiple access protocol," *Proceedings of IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 81–87, Mar 2005.

[84] S. Ray, J. B. Carruthers, and D. Starobinski, "RTS/CTS induced congestion in ad hoc wireless LANs," *In Proceedings of IEEE Wireless Communication and Networking Conference*, pp. 1516–1521, Mar 2003.

[85] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," *In Proceedings of International Conference on Mobile Computing and Networking*, pp. 61–69, Jan 2001.

[86] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," *In Proceedings of SPIE Conference on Wavelet Applications of Digital Image Processing*, pp. 376–387, Aug 1996.

[87] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Graceful degradation over packet erasure channels through forward error correction," *In Proceedings of Data Compression Conference*, Mar 1999.

[88] A. E. Mohr, R. E. Ladner, and E. A. Riskin, "Approximately optimal assignment for unequal loss protection," *Proceedings of IEEE International Conference on Communications*, vol. 1, pp. 367–370, Sep 2000.

[89] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Transactions in Multimedia*, vol. 6, no. 2, pp. 230–239, Apr 2004.

[90] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 165–168, Sep 2002.

[91] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, "Layered multiple description coding," *Proceedings of Packet Video Workshop*, Apr 2003.

[92] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Robust layered multiple description coding of scalable media data for multicast," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 154–157, Feb 2005.

[93] "Integrated services in the internet architecture: An overview," IETF RFC 1633, June 1994.

[94] R. Hamzaoui, V. Stankovic, and Z. Xiong, "Rate-based versus distortion-based optimal error protection of embedded codes," *Konstanzer Schriften in Mathematik*, no. 194, Oct 2003.

[95] T. H. Cormen, C. E. Leiverson, R. L. Rivest and C. Stein, *Introduction to algorithms*, 2nd ed. The MIT Press, 2001.

[96] Rensselaer Polytechnic Institute, "MC-EZBC Source code." [Online]. Available: http://www.cipr.rpi.edu/research/mcezbc/

[97] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzintsev, "Joint source-channel matching for a wireless communications link," *Proceedings of IEEE International Conference on Communications*, vol. 1, pp. 482–486, Jun 1998.

[98] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Tech. Journal*, no. 39, pp. 1253–1265, 1960.

[99] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Transactions on Communications*, vol. 43, no. 8, pp. 2316–2323, Aug 1995.

[100] T. Schierl, H. Schwarz, D. Marpe, and T. Wiegand, "Wireless broadcasting using the scalable extension of H.264/AVC," *Proceedings of IEEE International Conference on Multimedia Conference on Multimedia and Expo*, pp. 884–887, Jul 2005.

[101] R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control paradigm for video streaming in the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 18–32, Mar 2001.

[102] S. Johansen, A. Kim, and A. Perkis, "Quality incentive based congestion control for multimedia communication over IP networks," *Journal of Zheijang University / Springer-Verlag GmbH, Proceedings of Packet Video Workshop*, vol. 7, no. 1, pp. 7–12, Jan 2006.

[103] IETF, "An RTP payload format for erasure-resilient transmission of progressive multimedia streams," IETF Internet Draft: draft-ietf-avt-uxp-07.txt, April 2005.

[104] S. Gorinsky, S. Jain, and H. Vin, "Multicast congestion control with distrusted receivers," *Proceedings of NGC*, Oct 2002.

[105] V. Arya and T. Turletti, "Dealing with receiver misbehavior in multicast congestion control," Unité de recherche INRIA Sophia Antipolis, France, Research report no. 4899, 2003.

[106] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," ISO/IEC 8802-11:2005, Aug. 2005.

[107] IEEE 802.16j Relay task group, "Air interface for fixed and mobile broadband wireless access systems, Multihop Relay Specification — IEEE 802.16j," July 2007.

[108] IEEE P802.11 ESS mesh working group, "IEEE 802.11s draft D1.0," May 2007.

[109] T. Schierl, K. Gänger, C. Hellge, T. Stockhammer, and T. Wiegand, "SVC-based multi-source streaming for robust video transmission in mobile ad-hoc networks," *IEEE Wireless Communications Magazine*, vol. 13, no. 5, pp. 96–103, Oct 2006.

[110] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 235–246, Mar 2007.

[111] T. Stockhammer, T. Gasiba, W. A. Samad, T. Schierl, H. Jenkac, T. Wiegand, and W. Xu, "Nested harmonic broadcasting for scalable video over mobile datacast channels," *Wireless Communications and Mobile Computing: Special Issue on Video Communications for 4G Networks*, vol. 7, no. 2, pp. 235–256, Jan 2007.

[112] J.-P. Wagner, J. Chakareski, and P. Frossard, "Streaming of scalable video from multiple servers using rateless codes," *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 1501–1504, Jul 2006.

[113] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Microsoft Research technical report MSR-TR-2001-35, Feb. 2001.

[114] B. Cohen, "Incentives build robustness in BitTorrent," bittorrent.org, Tech. Rep., 2003.

[115] X. Zhang, J. Liu, B. Li, and Y.-S. P. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," *Proeedings of IEEE INFOCOM*, vol. 3, pp. 2102–2111, Mar 2005.

[116] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang, "A peer-to-peer network for live media streaming using a push-pull approach," *Proceedings of the ACM International Conference on Multimedia*, pp. 287–290, Nov 2005.

[117] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on selected areas in communications*, vol. 18, no. 6, pp. 819–828, Jun 2000.

[118] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan, "Application of FEC based multiple description coding to internet video streaming and multicast," *Proceedings of Packet Video workshop*, May 2000.

[119] T. Nguyen and A. Zakhor, "Distributed video streaming over the internet," *Proceedings of Packet Video workshop*, Apr 2002.

[120] X. Zhu and B. Girod, "Distributed rate allocation for video streaming over wireless networks with heterogeneous link speeds," *Proceedings of International Conference on Wireless Communications and Mobile Computing*, pp. 296–301, Sep 2007.

[121] Information Sciences Institute, University of Southern California, "The Network Simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/

[122] Joint Video Team (JVT), "JSVM software repository." [Online]. Available: jvtuser@garcon.ient.rwth-aachen.de:/cvs/jvt

[123] 3GPP, "3GPP TS 26.346 V6.10.0 technical specification group services and system aspects; multmedia broadcast/multicast service (MBMS); protocols and codecs," Sept. 2007.