Evaluating and Enhancing the Performance of IP-based Streaming Media Services and Applications

Odd Inge Hillestad

Centre For Quantifiable Quality of Service in Communication Systems
Centre of Excellence
Norwegian University of Science and Technology

A thesis submitted to the

Norwegian University of Science and Technology
Faculty of Informatics, Mathematics and Electrotechnics
Department of Electronics and Telecommunication

for the degree of PhD (philosophiae doctor)

May 2007

Abstract

This thesis deals with multimedia communication over unreliable and resource-constrained IP-based packet-switched networks. The focus is on estimating, evaluating and enhancing the quality of streaming media services, and streaming video services in particular.

The work can be divided into three parts. Part A covers issues related to predicting the perceived quality of streaming media applications. First, it presents a low-complexity method for estimating the amount of block-edge impairments in compressed video. The corresponding no-reference metric can be applied on the receiver side during a streaming session so as to enable automated objective video quality assessment, but is mostly relevant for MPEG-2 and early MPEG-4 (Part 2) systems.

Thereafter, part B presents an experimental multimedia testbed for measuring the performance, characteristics and error robustness of streaming media applications. An integral part of the test bed is the packet flow regenerator, which enables repeatable performance measurements of network components and streaming media clients. A performance study of this component is included. Further, using the real-time IP-network emulation capabilities of the testbed, the error robustness of a high-definition H.264/MPEG-4 AVC broadcast application is evaluated.

Finally, part C considers an adaptive streaming video-on-demand system where features of the upcoming H.264/MPEG-4 scalable video coding standard are combined with the use of an accumulation-based congestion control scheme in order to enable efficient video distribution over IEEE 802.16 wireless broadband networks.

Preface

This thesis 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 was conducted in the period January 2003 to February 2007. 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. Q2S is funded by the Norwegian Research Council, NTNU and Uninett. The PhD study was formally conducted at the Department of Electronics and Telecommunication, NTNU. Besides research work, it included compulsory courses corresponding to one semester of full-time studies, and one year of teaching assistance and related duties, which were kindly funded by the Department of Electronics and Telecommunications. In addition, from August 2005 to January 2006 I was a visiting scholar at the Performance Engineering Lab, University College Dublin (UCD), Ireland. This stay was funded by Q2S, NTNU.

I have collaborated closely with a number of people during my PhD studies. First, I would like to thank Professor Andrew Perkis at NTNU, who has been the supervisor of this work and provided encouraging support throughout my studies. Second, I would like to express my deepest gratitude to my co-authors and colleagues for our interesting discussions, their invaluable feedback and careful reviews. In particular, I would like to acknowledge and thank Dr. Ajit S. Bopardikar, Dr. Venkatesh Babu Radhakrishnan, Stian Johansen, Bjørnar Libæk and Dr. Ola Jetlund, together with Dr. Seán Murphy and Vasken Genc with whom I collaborated closely both during, and after my stay at UCD in Ireland. In addition, I would like to thank all my colleagues at Q2S and our MSc students for creating such a pleasant working environment. Finally, I'm sincerely grateful for all the care and support I continuously receive from those nearest to me, in particular my family and my dear Hanna.

Odd Inge Hillestad
Trondheim, Norway
May 2007

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Details
ACR	Absolute Category Rating
ADU	Application Data Unit
ARQ	Automatic Retransmission Request
ASO	Arbitrary Slice Ordering
AVC	Advanced Video Coding
CBR	Constant Bit Rate
CIF	Common Intermediate Format
CIP	Constrained Intra-Prediction
CPU	Central Processing Unit
DCCP	Datagram Congestion Control Protocol
DCT	Discrete Cosine Transform
DP	Data Partitioning
DRM	Digital Rights Management
FBWA	Fixed Broadband Wireless Access
FGS	Fine-Granular Scalability
FMO	Flexible Macroblock Ordering
FR	Full-Reference
FU-A	Fragmentation Unit - Type A
GOP	Group Of Pictures
GPS	Global Positioning System
HD	High Definition
HVS	Human Visual System

Abbreviation	Details
JM	Joint Model
JSVM	Joint Scalable Video Model
JVT	Joint Video Team (of ITU-T VCEG and ISO/IEC MPEG)
IDR	Instantaneous Decoding Refresh
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	IP Television
ITU-T	International Telecommunications Union - Telecommunications
MANE	Media Aware Network Element
MB	Macroblock (16 by 16 pixels)
MPEG	Motion Pictures Experts Group
MOS	Mean Opinion Score
MTU	Maximum Transmission Unit
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
NR	No-Reference
NSN	Next Sequence Number
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
PCAP	Packet Capture (Application Programming Interface)
PLR	Packet Loss Rate
PPS	Picture Parameter Set
PSS	Packet-switched Streaming Service
PSNR	Peak Signal to Noise Ratio

Abbreviation	Details
RFC	Request For Comments
RLE	Run-Length Encoding
RR	Reduced-Reference
RS	Redundant Slices
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SEI	Session Enhancement Information
SNR	Signal to Noise Ratio
SPS	Sequence Parameter Set
SS	Subscriber Station (in IEEE 802.16)
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
TFRC	TCP-Friendly Rate Control
UDP	User Datagram Protocol
UMA	Universal Multimedia Access
UME	Universal Multimedia Experience
VCEG	Video Coding Experts Group
VQEG	Video Quality Experts Group
VCL	Video Coding Layer
VoD	Video on Demand
VoIP	Voice over IP

Chapter 1

Introduction

This thesis considers delivery and communication of digital media over IP-based computer networks. More specifically, the focus is on evaluating and enhancing the quality of media services in resource-constrained unreliable network environments. This chapter motivates the research work (section 1.1), gives an outline of the thesis (section 1.2) and summarizes its main contributions (section 1.3). Publications are listed in section 1.4.

1.1 Motivation

Streaming media services delivered over IP (Internet Protocol) networks have been gaining serious momentum over the last years, and consumers show an increased interest in being able to play back, and enjoy their media wherever they are. There are several examples from contemporary media and entertainment industry that clearly indicate the future potential for such services.

Firstly, the success and large-scale deployment of portable media players like the Apple iPod [1] suggests a high consumer demand for portability and mobility; users will increasingly use media services in new contexts and scenarios. Secondly, the recent emergence of hugely popular video portals like YouTube [2], which are largely based on user-contributed video content, shows a shift in people's relation to digital media; besides being traditional media consumers, end-users have started creating, distributing and sharing their media. While today's music and video services are mostly based on a download and play service model, it is anticipated that more sophisticated and interactive multimedia services based on real-time delivery will emerge in the future. As a last example, IPTV – IP-based distribution of television channels together with video-on-demand – has recently been the subject of much interest in the industry, as Internet service providers

try to make the most of their investments in high capacity infrastructure.

However, this evolution presents new challenges; ubiquitous usage of IP-based multimedia services requires transparent delivery of media resources to end-users irrespective of network access, type of connectivity, or current network conditions. In addition, end-users may have communication devices with widely different and limiting capabilities, and they may also have different preferences with respect to how the multimedia content is presented to them. Future multimedia systems and applications need to take these considerations into account, and be able to adapt properly. Some essential aspects of these problems are being addressed through the concept of Universal Multimedia Access (UMA) [3].

There are many technological factors that have enabled or facilitated the delivery and usage of higher quality, more appealing networked multimedia services. Two important ones are the ever-increasing deployment of higher bandwidth wired and wireless networks, and the continuous development of more efficient compression schemes for audio and video. Common for both of these – and indeed for most aspects of technological innovation in general – is the importance of standards; they facilitate universal deployment of interoperable services, and enable different companies to produce devices that work seamlessly together. This stimulates innovation and competition, which is essential for economic growth.

The Internet has made IP-based distribution the preferred choice when deploying new services. The simple and flexible design of the Internet protocols was important for their success and popularity. Having most of the intelligence in the endpoints made it easy to deploy and interconnect networks, but this had clear implications on the amount of functionality that could be provided to end-user applications [4]. More specifically, the traditional best-effort service model can not provide any service guarantees with respect to application throughput, packet loss, delay and delay jitter. Unpredictable delay jitter is destructive as it may translate directly to packet loss for real-time applications. Thus, the lack of network Quality of Service (QoS) makes it exceedingly hard to provide reliable multimedia services with a certain quality to end-users.

This brings us to another important challenge, which is related to the end-user perceived quality of the delivered service. While the objective of UMA enabled systems is to provide the user with the best possible subset of a multimedia resource that the user is capable of receiving, the aggregate end-to-end quality as perceived by the end-user is often referred to as *Quality of Experience* (QoE) [5]. However, to measure the end-user perception of an audiovisual service quality, or similarly, to which extent they react objectionable to distortions introduced by compression and packet-switched transmission, is very difficult and depends on factors that are not easily modeled. Examples of such factors are; context in which the service is being used, user expectation, human diversity, preferences,

and application knowledge.

Finally, a necessary prerequisite for evaluating the quality of a service is to have appropriate facilities for doing performance evaluations. For instance, a study may want to investigate the quality of a multimedia service – as perceived by an end-user – when the service is operating under certain network conditions. Decisions then have to be made regarding an appropriate test setup (in addition to usage context and other assumptions). Traditionally, computer simulations using models of application behavior have been used. However, making good simulation models is a difficult task [6]. Experiments that are more realistic can be performed using real-world applications, deployed in local or wide area test environments. However, it is difficult to combine the realism of a testbed – consisting of real (prototype) applications and networking devices – with the control and repeatability offered by computer simulations.

To summarize, there is a trend towards embracing user aspects of perceived quality into the design and optimization of networked media services. New tools and methods are needed to evaluate and predict end-user perceived quality under normal operating conditions of a service/application. The first work presented in this thesis considers some initial aspects of this challenge, more specifically related to typical visual impairments that are introduced by video compression in such systems. Further, the thesis considers the problem of how to systematically and realistically evaluate the performance of a media service/application – under specific network operating conditions – in a controlled and repeatable manner. In addition, this work explores the design and evaluation of a standards-based video-on-demand solution for distribution over wireless broadband access networks. The video streaming system should be able to adapt its transmission rate based on the current network conditions and level of congestion on the wireless access link. It is important that the system works in such a way as to ensure good and smooth video quality, and prevent users from experiencing interruptions in continuous playback. Such a system would have the potential of providing end-users with a greatly enhanced QoE.

1.2 Outline of the Thesis

This thesis is divided into three main parts. First, part A deals with estimation of video quality at the receiver side of a multimedia communication system. Such information could be very helpful in deciding how well such a system works in a real-world deployment. This again could be used to tune and enhance the operation of the system.

Secondly, part B considers how such a multimedia communication system could be evaluated in a laboratory setting, before it actually is deployed. Testing and performance evaluation needs to be done in a controlled and repeatable

manner, and results from such experiments can be used to optimize system configuration. Towards this end, a multimedia testbed is presented that allows transmission, measurement, capture and regeneration of media streams in a controlled network environment. The testbed is used to evaluate the performance of a high-quality high-definition (HD) H.264/AVC broadcast video service when packet loss occurs in the network.

Finally, part C presents a standards-based application and system study; an adaptive video-on-demand solution based on H.264/MPEG-4 AVC scalable video coding (SVC) and a media-friendly congestion control is presented, which is shown to enable efficient video distribution over IEEE 802.16 fixed broadband wireless networks. A prototype of the video streaming system is presented, and its performance is evaluated using a simulated 802.16 network model, and the testbed presented in Part B.

At this point, an important remark has to be made. When this research work was performed, a technology for facilitating fine-granular scalability (FGS) in video coding was an integral part of the H.264/MPEG-4 scalable video coding (SVC) amendment. However, during the final stages of writing this thesis, a decision was made by the standardization body to remove FGS from SVC, and possibly include this technology in a later amendment to the H.264/AVC standard [7]. In the following, since the work described here was performed prior to this decision, please consider FGS as a possible future extension to H.264/MPEG-4 AVC and SVC.

1.3 Main Contributions

This section summarizes the most important results, and the main contributions of this thesis.

The main contribution of part A is;

- A simple, low-complexity method for estimating block-edge impairments in compressed video. The method does not utilize the original video signal for predicting these impairments. Hence, it may be used to predict the perceived impairment of block-edge artifacts at the receiver side of a visual communication system.

The main contributions of part B are;

- Design and development of a testbed for evaluating the end-to-end performance of streaming media applications. The testbed consists of application and network components; most notably, the combination of a high-performance packet monitoring and capture device, and a packet flow

regenerator, allows repeatable experiments using real-life streaming media applications and network devices.

- A performance evaluation of the packet flow regenerator showed that media streams may be accurately replayed at several Mb/s. For instance, in the range of 5-20 Mb/s, over 99 % of packet interarrival times are off by less than 2 ms.

- Using the testbed described above, the error robustness of a high-definition H.264/AVC broadcast service was evaluated. The video service was subjected to uniformly distributed packet loss using a real-time network emulation device, which showed the effectiveness of some error resilience tools in the H.264/AVC standard such as slice partitioning and flexible macroblock ordering.

The main contributions of part C are;

- A simulation study of video distribution to rural areas over IEEE 802.16 broadband wireless access networks. In a realistic simulation scenario, if 802.16 coverage area is preferred over system efficiency (more spectral-efficient coding and modulation), it was shown that such a system may provide around 7 Mb/s of bandwidth to video applications, corresponding to ten simultaneous users streaming videos delivered at 700 Kb/s on average.

- An video streaming system based on H.264/MPEG-4 Scalable Video Coding that adapts the amount of enhancement layer information transmitted based on the occupancy of the receiver playout buffer. This method was shown to give end-users similar risk of experiencing interruption in continuous media playback due to buffer underflow.

- A standards-based adaptive video streaming system that combines H.264/MPEG-4 Scalable Video Coding with a rate adaptation algorithm derived from accumulation-based congestion control. The proposed solution was shown to give high utilization of the wireless link resources, while providing fair and smooth transfer rates to all video applications. At the same time, buffer underflows were effectively prevented.

1.4 Publications

This thesis is based on work that has been published in a number of papers, which are listed below.

Refereed International Conference Publications

- Odd Inge Hillestad, Venkatesh Babu Radhakrishnan, Ajit S. Bopardikar and Andrew Perkis, "Video Quality Evaluation for UMA", in *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004)*, Lisboa, Portugal, April 21–23, 2004.

- Venkatesh Babu Radhakrishnan, Ajit S. Bopardikar, Andrew Perkis and Odd Inge Hillestad, "A No-reference Measure for the Effect of Packet-loss in Video Streaming over IP", in *Proceedings of the 11th International Workshop on Systems, Signals and Image Processing (IWSSIP 2004)*, Poznan, Poland, September 13–15, 2004[1].

- Ajit S. Bopardikar, Odd Inge Hillestad and Andrew Perkis, "A Temporal Error Concealment Algorithm Based on Structural Alignment for Packet Video", in *Proceedings of the 7th IEEE International Conference on Signal Processing and Communications (SPCOM 2004)*, pp. 373–377, Bangalore, India, December 11–14, 2004[2].

- Venkatesh Babu Radhakrishnan, Ajit S. Bopardikar, Andrew Perkis and Odd Inge Hillestad, "No-reference Metrics for Video Streaming Applications", in *Proceedings of the 14th International Packet Video Workshop (PV 2004)*, Irvine, CA, USA, December 13–14, 2004.[3].

- Andrew Perkis, Peder Drege and Odd Inge Hillestad, "UMA Enabled Environment for Mobile Media Using MPEG-21 Client and Server Technology", in *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005)*, Montreux, Switzerland, April 13–15 2005[4].

- Ajit S. Bopardikar, Odd Inge Hillestad and Andrew Perkis, "Temporal Concealment of Packet Loss Related Distortions in Video based on Structural Alignment", in *Proceedings of the Eurescom Summit 2005*, Heidelberg, Germany, April 27–29, 2005.

- Odd Inge Hillestad, Bjørnar Libæk and Andrew Perkis, "Performance Evaluation of Multimedia Services Over IP Networks", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005)*, pp. 1464–1467, Amsterdam, The Netherlands, July 6–8, 2005.

[1]The author (OIH) contributed to the final writing of the paper.

[2]The author (OIH) contributed to the development of the error concealment algorithm described in this paper, and performed simulations evaluating its performance.

[3]The author (OIH) contributed to the final writing of the paper, and to the development of the NR block-edge impairment metric described in this paper.

[4]The author (OIH) contributed to the final writing of the paper.

- Andrew Perkis, Solveig Munkeby, and Odd Inge Hillestad, "A Model for Measuring Quality of Experience", in *Proceedings of the 7th Nordic Signal Processing Symposium (NORSIG 2006)*, Reykjavik, Iceland, June 7–9 2006[5].

- Odd Inge Hillestad, Andrew Perkis, Vasken Genc, Seán Murphy, and John Murphy, "Delivery of On-Demand Video Services in Rural Areas via IEEE 802.16 Broadband Wireless Access Networks", in *Proceedings of the 2nd ACM International Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP 2006)*, volume 1, pages 43–51, Hussein Alnuweiri and Regina Araujo (editors), Torremolinos, Malaga, Spain, October 2–6, 2006. ACM Press.

Refereed International Journal Publications

- Odd Inge Hillestad, Ola Jetlund, and Andrew Perkis, "RTP-based Broadcast Streaming of High Definition H.264/AVC Video: An Error Robustness Evaluation", Journal of Zhejiang University - Science A, 7(0):19–26, 2006. Presented at the International Packet Video Workshop (PV 2006), Hangzhou, China, 2006.

- Venkatesh Babu Radhakrishnan, Andrew Perkis, and Odd Inge Hillestad, "Evaluation and Monitoring of Video Quality For UMA Enabled Video Streaming Systems", *Multimedia Tools and Applications*, accepted for publication, 2007, Springer Netherlands[6].

Non-refereed National Journal and Conference Publications

- Odd Inge Hillestad, Ola Jetlund and Andrew Perkis, "Error Robustness Evaluation of High Quality H.264/AVC Broadcast Services over RTP/IP Using Network Emulation", presented at *Norsk Nettforskningsseminar*, October 27–28 2005.

- Andrew Perkis, Peter Svensson, Odd Inge Hillestad, Stian Johansen, Jijun Zhang, Asbjørn Sæbø, and Ola Jetlund, "Multimedia over IP Networks", *Telektronikk*, 1(Real-time communication over IP):43–53, 2006[7].

[5] The paper summarizes the results of Munkeby's M.Sc. thesis. The author was assistant supervisor for this work.

[6] The author (OIH) contributed to the final writing of the paper, and to the development of the NR block-edge impairment metric described in this article.

[7] The author contributed to the literature review in this overview article.

Chapter 2

Background

This chapter will introduce some important aspects of multimedia distribution over IP-based networks. Section 2.1 presents standardized formats for representing multimedia content, in particular the MPEG-4 standard and its state-of-the-art video codec; Advanced Video Coding (AVC). Then, section 2.2 will present and discuss issues related to the delivery of such media over packet-switched IP networks.

2.1 MPEG-4: the Multimedia Standard

MPEG-4 is an ISO/IEC standard for representing interactive, rich multimedia presentations [8]. It was developed by the Motion Pictures Experts Group (MPEG). The term *rich* refers to its capabilities of representing different types of media; visual and audible, natural and synthetic/animated, 2D and 3D, regularly and arbitrarily shaped. MPEG-4 is a massive standard, and currently consists of 22 separate parts. Fundamental concepts related to the composition and representation of multimedia presentations – e.g. the organization of media objects in a scene description, and how placeholder object descriptors facilitate tight synchronization of media objects – are described in Systems (part 1). Other important characteristics of the MPEG-4 system model are that the design allows dynamic and streamed scene descriptions, and the potential for protecting media streams independently using digital rights management (DRM) technology.

The actual coded representation of media objects – elementary streams – are defined in other parts of MPEG-4, like Visual (part 2), Advanced Video Coding (part 10), and Audio (part 3). Delivery of MPEG-4 content can be performed over literally any transport medium thanks to the protocol-independent design and the Delivery Multimedia Integration Framework (DMIF - part 6). Further, file

format specifications describe how scene descriptions, media object descriptions and corresponding elementary streams can be wrapped together into an MP4 file.

The work described in this dissertation only utilizes a small subset of the technologies within the MPEG-4 standard, namely the specifications related to coding of natural video and the MP4 file format. Also, note at this point that subsets of the tools and technologies specified in the various parts of the MPEG-4 standard are combined into *profiles*, which target specific applications and serve as conformance points for equipment and software vendors. Further, a profile may specify different *levels* that reflect the complexity of the system, e.g. by placing constraints on the maximum bitrate and picture size for video. For MPEG-4 part 2 video, profiles include e.g. the simple profile (SP) and the advanced simple profile (ASP). Correspondingly, H.264/MPEG-4 AVC profiles include the baseline, extended, main and high profile.

2.1.1 H.264/MPEG-4 Advanced Video Coding

H.264/MPEG-4 AVC (Advanced Video Coding) is the current state-of-the-art international video coding standard. It was developed in a collaborative effort known as JVT (Joint Video Team) between MPEG of ISO/IEC [9] and the Video Coding Experts Group (VCEG) of ITU-T [10]. Throughout this text it will be referred to as H.264/AVC. There are several excellent overview papers covering the H.264/AVC standard; most notably [11], [12], and [13]

The development of H.264/AVC [14] marked an important milestone in the development of video compression technologies. The standard was specifically designed for delivery over packet-switched networks, and has been reported to give over 50% gains in coding efficiency compared to that of MPEG-2 [12]. While the standard includes a multitude of new and improved features, some basic concepts are unchanged; similar to previous ITU-T and MPEG video coding standards, H.264/AVC is a block-based motion-compensated hybrid video coding scheme. The significant improvements in compression efficiency are made possible through enhanced prediction techniques and improved entropy coding. In addition, an error resilient design and a considerable amount of tools to increase robustness to packet loss allows for efficient transmission and improved quality in error-prone environments.

The standard defines two main layers; a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). The definition of the VCL includes tools and methods to code a set of macroblocks (MB) of a picture into a slice partition or a data partition. Here, a coded *picture* can represent both a progressively scanned frame and an interlaced field. The slice/data partitions are conceptually transferred by the network abstraction layer as NAL units, which are the basic transport units. Essential picture header information is assembled in

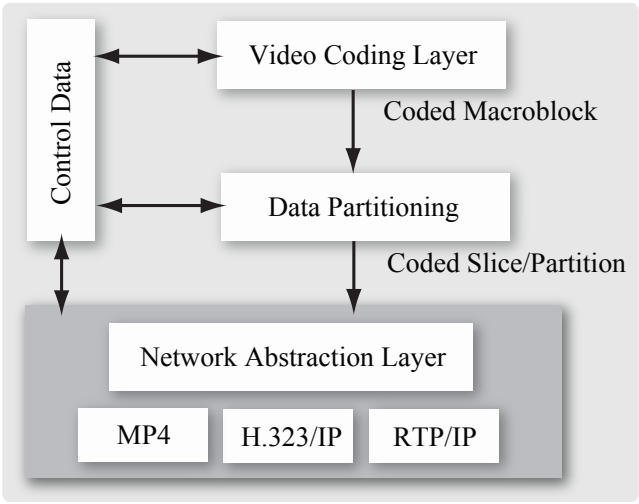


Figure 2.1: High level architecture of H.264/AVC.

parameters sets, more specifically the Sequence Parameter Set (SPS) and Picture Parameter Sets (PPS). Other control data include messages that may control and enhance the operation of the decoder during a session – so-called Supplemental Enhancement Information (SEI) NAL units.

The Video Coding Layer

The video coding layer specifies how to decode of a set of coded macroblocks contained in a slice partition or data partition. Among the new tools in the VCL, some provide gains in compression efficiency through improved and more flexible prediction. While bi-directional temporal prediction is well-known from previous coding standards, H.264/AVC gives an encoder more freedom and flexibility in choosing reference pictures. Two design aspects related to this are (1) the decoupling of the orders in which picture are referenced and displayed and (2) the decoupling of type of picture and its ability to be used for reference [11]. For instance, B pictures in MPEG-2 have to use the previous and next I/P pictures as reference and they can not themselves be used as reference. In contrast, H.264/AVC places no specific restrictions on the order and type of picture being used for prediction. Rather, it places a restriction on the size of the reference picture buffer in terms of memory footprint, and the maximum number of pictures in this buffer. This ensures that all decoders conforming to a given profile and level can decode the stream.

This concept of generalized B pictures [15], together with the use of explicit commands that control the operation of the reference picture buffer (and reference picture lists), allows an H.264/AVC encoder to build arbitrary explicit coding structures. Hierarchical coding structures have been shown to have some very advantageous properties [16]. Besides potentially giving a higher compression efficiency compared to traditional coding structures like e.g. `I-B-B-P...` , the hierarchical structure inherently provides temporal scalability. Figure 2.2 shows a hierarchical coding structure with a group of pictures (GOP) size of 8 pictures, which enables temporal reconstruction at four different levels. Note that in order to reconstruct the bitstream at a particular temporal level, all temporal levels up to and including the chosen level have to be decoded.

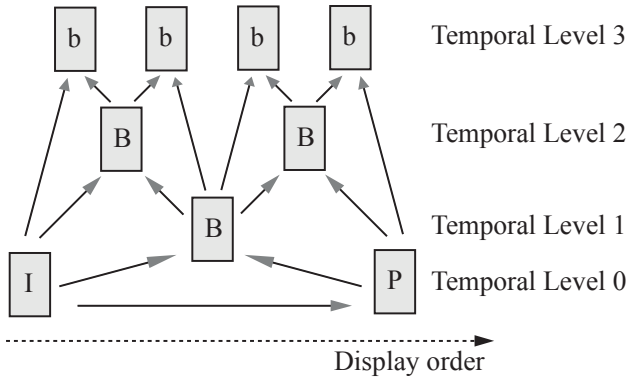


Figure 2.2: Hierarchical coding structure with four temporal scalability levels.

On the macroblock level, some of the techniques that improve the prediction capabilities of H.264/AVC are: variable block size motion compensation with quarter-pixel motion vectors, more efficient direct coding modes, and enhanced intra prediction. Block-edge artifacts, which have been common to block-based coding schemes, are less prominent in H.264/AVC due to a mandatory adaptive de-blocking filter within the motion compensation loop. While this adds some complexity in the decoder, the reduction, and often complete elimination of block-edge artifacts significantly improves subjective quality. However, the smoothing effect of the in-loop de-blocking filter means that H.264/AVC video suffers more from blurriness at low bit rates.

Other tools in the VCL provide gains in compression efficiency through more efficient representation. The samples of intra-coded macroblocks (possibly after intra-prediction), or the residual in the case of P and B slices, is transformed using a 4x4 integer transform (or optionally an 8x8 transform in the High Profile). This reflects the variable block-size motion compensation, allows implementation

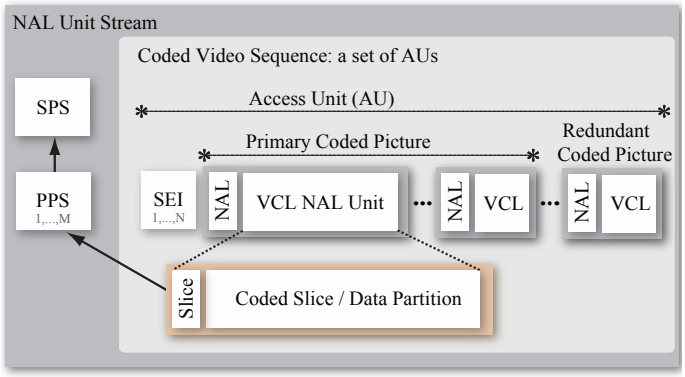


Figure 2.3: Packetization of H.264/AVC streams.

on 16-bit processors, and eliminates the drift problem caused by the non-exact inverse Discrete Cosine Transform (DCT) transform. After quantization, the transform coefficients are encoded using context adaptive entropy coding, either variable-length (CAVLC) or binary arithmetic coding (CABAC) [11].

The Network Abstraction Layer

The structure of a typical packetized H.264/AVC bitstream is illustrated in Figure 2.3 (the terminology is explained by Wiegand et al. in [11]). A NAL unit stream may consist of one or more coded video sequences, which in turn always start with an Instantaneous Decoding Refresh (IDR) NAL unit. Since an IDR NAL unit resets all prediction and state in the reference picture buffer, a coded video sequence is independently decodable. Further, a coded video sequence consists of a set of access units. The decoding of an access unit results in a decoded picture (frame or field).

The network abstraction layer provides a clean interface to the different types of coded video data and control data. From a transport point of view, all data generated by the video encoder are NAL units of different type and importance. In this way, it facilitates the delivery of H.264/AVC over a variety of transport layers, as can be seen in Figure 2.1.

A NAL unit has a one-byte header, as shown in Figure 2.4[1]. The type of data is signalled in a specific 5-bit field called `nal_unit_type`. The relative importance of NAL units is signalled in the 2-bit `nal_ref_idc` field, which hence can take

[1] "fzb" is an abbreviation for forbidden zero bit, and must be equal to zero for a conforming stream.

Figure 2.4: The NAL Unit Header.

on values from 0 to 3. Higher values indicate higher importance; e.g. if a NAL unit contains VCL data, and has a non-zero `nal_ref_idx` value, the slice or slice partition contained in that NAL unit is part of a reference picture. This design enables streaming servers or so-called media-aware network elements (MANE) inside the network to easily find out what packets are most important, if e.g. some NAL units have to be skipped to reduce the rate of a stream. The format of NAL units is identical in networked and non-networked applications. However, when a video sequence is stored in an MP4 file or transmitted in byte-oriented systems, the NAL units are preceded by a 3-bytes start code. This is defined in Annex B of the standard [10], and it is therefore referred to as the Annex B byte stream format. In packet-switched systems like RTP, the NAL unit is mapped directly to packet payload (this will be revisited in section 2.2.2).

Error Resilience in AVC

Since packet-switched delivery in error-prone environments was one of the application areas specifically targeted by H.264/AVC, its architecture was designed to be resilient against transmission errors. As noted above, essential header information is gathered in parameter set NAL units. Syntax elements common to the entire sequence are assembled in the *sequence parameter set* (SPS), while elements common to one or more pictures in the sequence can be found in *picture parameter sets* (PPS). Because of their importance, they would typically be transported in a reliable out-of-band fashion. If the application and delivery scenario does not allow this, parameter sets could be repeated to increase system robustness.

In addition to an error resilient design, H.264/AVC provides several tools to increase the robustness to packet loss. The partitioning of a picture into slices is very flexible, both in terms of slice sizes and with respect to which macroblocks are allocated to each slice. In addition to simply assigning the macroblocks of a picture to slices in raster scan order, a scheme known as flexible macroblock ordering (FMO) can be used to partition a picture into a number of slice groups using a macroblock allocation map [17]. Pre-defined allocations like e.g. Interleaved and Dispersed slice groups are specified, but the standard also enables explicit allocation of macroblocks to a slice group. Such explicit macroblock allocation maps, together with features that limit in-picture or inter-

picture prediction, enable the design of isolated regions [18]. This allows an encoder to give better protection to high-importance regions-of-interest within a video sequence.

Other error resilience tools include arbitrary slice ordering (ASO), data partitioning (DP), and redundant slices (RS). Because the coding mode (e.g. intra, inter, skip) can be decided on a macroblock level, the standard supports insertion of intra-coded macroblocks into P or B pictures to stop error propagation [19]. For more information on the error resilience features of H.264/AVC and transport in error-prone environments see [20], [21] and [17]. Table 2.1 summarizes some of the error resilience tools available in H.264/AVC, and which tools that can be used in the different profiles of the standard.

Profile:	*Baseline*	*Extended*	*Main*	*High*
Arbitrary Slice Ordering (ASO)	yes	yes	no	no
Constrained Intra Prediction (CIP)	yes	yes	yes	yes
Flexible Macroblock Ordering (FMO)	yes	yes	no	no
Maximum Slice Groups	8	8	1	1
Intra & IDR pictures	yes	yes	yes	yes
Intra MB refresh	yes	yes	yes	yes
Multiple Reference Pictures	yes	yes	yes	yes
Parameter Sets	yes	yes	yes	yes
Repetition of SPS & PPS	yes	yes	yes	yes
Recovery Point SEI message	yes	yes	yes	yes
Redundant Slices (RS)	yes	yes	no	no

Table 2.1: H.264/AVC profiles and error resilience tools

2.1.2 The Scalable Video Coding Extension

UMA-enabled applications require that multimedia content can be delivered and shown on a variety of devices with different network connectivity and capabilities e.g. regarding screen size and processing power. In addition, as previously mentioned, packet-switched delivery require elastic, rate-adaptive applications. Within such an adaptive framework, scalable video coding is a natural approach [22].

Therefore, of particular interest to future adaptive streaming media applications is the current standardization effort within JVT concerning a scalable extension of H.264/AVC [23, 24]. This amendment is known as SVC (Scalable Video Coding), and will specify how to represent video streams that enable spatial, temporal and quality scalability. More specifically, SVC will define how

to represent *enhancement coded pictures* that may augment the primary coded pictures of an H.264/AVC stream. In this respect, it can provide supplementary *enhancement layers* on top an H.264/AVC compatible *base layer*.

A stream is considered to be scalable if a subset of the full bitstream can be extracted to produce a valid conforming stream. Compared to the full stream, the sub-stream may have a lower spatial resolution, lower temporal resolution, and/or lower visual quality in terms of SNR. A given reconstruction point is identified by the triplet S_{Id}, T_{Id} and Q_{Id}, corresponding to spatial dependency layer, temporal level and quality layer, respectively. Further, it is possible to combine different scalability modes.

Temporal Scalability

The temporal reconstruction points are typically an integer fraction of the maximum frame or field rate. Using a dyadic temporal decomposition structure of hierarchical B pictures (as exemplified in Figure 2.2), this fraction is equal to 2. Hence, if the full frame rate of the video sequence is 30 frames/s, it is possible to select a subset of NAL units that allow reconstruction at 15, 7 and 3.5 frames/s. The pictures in between two consecutive pictures of the base temporal level T_0 is called a group of pictures (GOP), or a sub-stream. Readers are referred to [23] for a discussion on other coding structures that provide temporal scalability, e.g. non-dyadic and low-delay variants. It should be mentioned that to increase compression efficiency of hierarchical coding structures, a coarser quantization is applied in the pictures of higher temporal levels as compared to the pictures in level T_0. This leads to rather significant fluctuations in terms of peak-signal-to-noise-ratio (PSNR), as will be discussed later in section C.3.

Spatial Scalability

Spatial scalability is achieved through a layered approach with a downsampling stage, parallel encoding of each spatial layer, and optional use of inter-layer prediction techniques (including prediction of motion vectors and residual from lower layers). By adding some constraints on which macroblocks that can be used for intra prediction between layers, single-loop decoding of each spatial layer is possible. The spatial base layer has a dependency identifier $D_{Id} = 0$. If more than one spatial layer is used, they are coded simultaneously, and transmitted in order of increasing D_{Id}.

Quality Scalability

Before describing how quality scalability is facilitated in SVC, an important remark has to be made. When this research work was performed, both fine-

granular scalability (FGS) and coarse-grained scalability (CGS) were integral parts of SVC. However, during the final stages of writing this thesis, a decision was made by the JVT to remove FGS from the set of tools included in the forthcoming SVC amendment [7]. Apparently, the complexity associated with FGS decoding was considered too high compared to that of single-layer coding and CGS, and more research needs to be performed in order to address this problem. In the following, since the work described here was performed prior to this decision, please consider FGS as a possible future extension to H.264/AVC and SVC.

In the case of SNR quality scalability, there are two distinct types in SVC. First, a layered approach using inter-layer prediction techniques leads to coarse-grained scalability (CGS). This allows reconstruction at a finite set of rate points, but coding efficiency tend to decrease when the distance between rate points are smaller. To enable more flexible rate adaptation, SVC also supports fine-granular scalability (FGS) through so-called *progressive refinement* (PR) slices. Refinement of the residual signal is done through an embedded quantization technique applied to transform coefficients of different quality layers. The quantization parameters (QP) used in quality enhancement layers are determined from the QP specified for the base quality layer. More specifically, the standard specifies an increase of 6 parameter units per quality layer, which corresponds to a reduction of the quantization step size with a factor of two. As its name implies, a PR slice can be truncated at an arbitrary point[2]. The feature is enabled by cyclic scanning of transform coefficient from the macroblocks of the picture, and level of granularity can be traded off against complexity by varying the number of cycles. SVC also supports refinement of motion vectors. Note also that PR slices can be divided into several NAL units [25].

To obtain the highest coding efficiency for quality enhancement layers, prediction in the temporal domain should be based on the enhancement layer reconstructed pictures. However, during transmission and adaptation, if some quality enhancement layers are truncated or dropped, this may lead to a reconstruction mismatch. This mismatch propagates until the next intra-coded picture, and is known as the drift problem. In SVC, the enhancement layer reconstructed picture is by default used for temporal prediction, unless the picture is explicitly marked otherwise. In a hierarchical coding structure, drift can be limited to within a GOP if the predictively coded key pictures at temporal level T_0 is marked in this way. This has significant benefits in terms of coding efficiency for SNR scalable streams [23].

In addition, the standard includes a concept known as leaky prediction to handle the drift problem in an explicit way. The key pictures in T_0 may reference a weighted average of the base and enhancement layer reconstruction of other

[2]Truncation of progressive refinement slices needs to be byte-aligned

pictures. This weight can be adaptively chosen to optimize the trade-off between coding efficiency and drift. Leaky prediction is especially suited for low-delay applications in which IPPP coding structures is preferred to the hierarchical structures.

SVC Transport Interface

To facilitate transport and flexible adaptation of a scalable SVC stream, some basic information about the properties of, and dependencies between SVC slices are contained in a specific SVC NAL unit header. This three-byte header immediately follows the general H.264/AVC NAL unit header if the NAL unit type is equal to 20 or 21. In addition to the triplet D_{Id}, D_{Id}, and D_{Id}, it specifies whether the NAL unit is required for inter-layer prediction and whether it can be truncated or not. The relative priority of the SVC NAL units – a very useful piece of information for doing adaptation within the network itself – is signalled in a priority identifier P.

To increase a decoders resilience to loss of the all-important base layer pictures, a mechanism was adopted in which enhancement layer pictures reference which key picture they depend on using a frame number. This existence of a 1-byte frame number is signalled in the SVC NAL unit header for NAL units having DId=0 and QId=0 (base layer NAL units).

For more information on the transport interface of SVC, the reader is referred to [26].

2.2 Streaming Multimedia over IP networks

In [27], 3GPP describes streaming as referring to "the ability of an application to play synchronized media streams like audio and video in a continuous way while those streams are being transmitted to the client over a data network".

Figure 2.5 depicts a typical streaming session in which some pre-encoded media content is being delivered from a streaming server. Compressed media data is wrapped in RTP, UDP and IP headers, and transmitted over an access network. On the network layer, the media data is only recognized as a packet flow throughout the session. Because IP networks introduce packet loss, delay and delay jitter, a playout buffer is needed to both absorb the variation in delay and allow for retransmission of dropped packets. To alleviate the effect of missing media data, the decoder should be error resilient and be able to perform error concealment [28] [29].

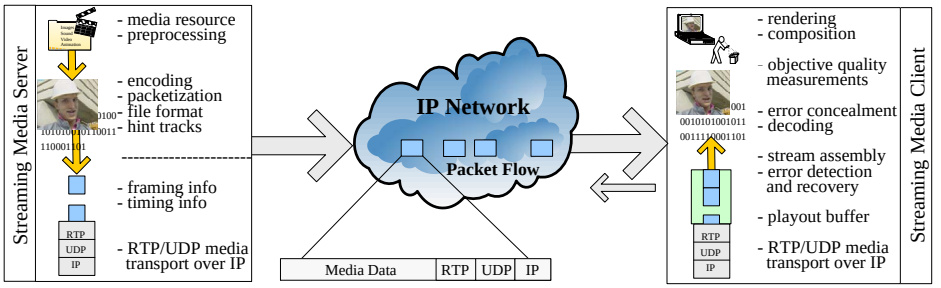


Figure 2.5: A model of a streaming system

2.2.1 Requirements of Multimedia Applications

In general, different multimedia applications have different requirements with respect to the quality of service delivered by the underlying transport network. For instance, the delay requirements for two-way interactive services are much more stringent than for video-on-demand services (VoD), and IP television broadcast (IPTV) [30]. However, some generalizations can be made; two important requirements for IP-based streaming media applications are 1) real-time delivery to prevent buffer-underflow events, and 2) smooth rate changes to prevent oscillations in perceived quality. Further, while video applications can be designed to have great resilience against packet loss (see section 2.2.4), it would be advantageous to minimize the error rate when the system is operating in steady state. At the same time, proper congestion control and avoidance mechanisms are essential for the efficiency and stability of the network, and for making sure that system resources are divided fairly between applications and end-users. These issues will be further discussed in section 2.2.3.

2.2.2 Standard Protocols for Streaming Media

The real-time constraint is typically a more important requirement than reliable in-order transfer for multimedia applications. Therefore, the connectionless unreliable UDP protocol is usually preferred over TCP, particularly for highly delay-sensitive applications like video conferencing and voice over IP (VoIP). To enable features such as loss detection, error reporting and proper synchronization of media streams, a more sophisticated protocol is needed on top of UDP.

IETF protocols for audiovisual transport

The Real-Time Protocol (RTP) is the IETF (Internet Engineering Task Force) standard transport protocol for real-time media, and supports both unicast and

multicast applications [31]. The term unicast describes a typical point-to-point delivery scenario, as exemplified by the client-server session in Figure 2.5, whereas multicast covers point-to-multipoint and multipoint-to-multipoint delivery.

Note that RTP may also be used together with newer protocols like the Datagram Congestion Control Protocol (DCCP), which provides an unreliable connection-oriented service [32].

In RTP, all media data is wrapped in a fixed 12 byte RTP header, consisting of a total of 9 fields. The maximum size of an RTP packet is 65536 bytes, minus IP/UDP headers, but should be restricted to the maximum transmission unit (MTU) size of the underlying network. An RTP receiver can detect lost packets and perform packet reordering through the use of sequence numbers, which increase by 1 for every RTP packet that is transmitted. The sampling instant associated with all access units (video frames/fields) can be conveyed in a 32 bit time stamp. For video applications, a 90 KHz clock is typically used, which implies that the RTP time stamp would increase by 3000 for every frame of a 30 frames/s progressive video source. Boundaries between video frames can be identified by the 1-bit marker field, which is usually set for the very last RTP packet of an access unit. As RTP was intended to be used in multiparty communication, all RTP senders must have a unique identifier – an SSRC (Synchronization Source Identifier).

To ensure that new and future media formats easily can be deployed over RTP, the protocol is designed to be media-independent. The type of media being transported, e.g. H.264/AVC video, is specified in the payload type field of the RTP header. Description on how to packetize and deliver e.g. H.264/AVC video over RTP is given in an accompanying payload format specification [33], and will be covered later in section 2.2.2.

An RTP flow is associated with a corresponding RTCP control channel. The control channel can be used to report link statistics or provide sender information to one or several receivers. RTCP sender reports are amongst other things used in long-term synchronization, and estimation of sending rate and round-trip time. On the other hand, RTCP receiver reports include feedback related to quality of service (QoS) such as short-term and long-term packet loss, and interarrival jitter. More comprehensive receiver reports are also possible using RTCP eXtended Reports (XR) [34]; elements include e.g. more detailed loss and reception statistics, together with the current delay experienced in the playout jitter buffer. While [34] is somewhat geared towards VoIP applications, it provides a common framework for future RTCP extensions.

A streaming session is typically started, initiated and controlled using RTSP (Real-Time Streaming Protocol [35]) or SIP (Session Initiation Protocol [36]). RTSP messages include SETUP, PLAY, PAUSE, STOP and BREAKDOWN. SIP is used in videoconferencing applications. When a new session is initiated,

SDP (Session Description Protocol) is used in both RTSP and SIP to describe the session and communicate the type and location of the RTP media streams that constitute a multimedia presentation [37]. For each media stream, one line of text identifies the media type (e.g. video or audio) contained in an RTP session. Another line maps that media session to a specific payload type and RTP time stamp frequency. The SDP description may also include elements that configure receiving decoders, such as H.264/AVC parameter sets.

RTP payload format for H.264/MPEG-4 AVC

RFC 3984 [33] specifies the mapping of NAL units to RTP packets and describes issues related to fragmentation and aggregation of NAL units. Three different packetization modes are possible; first, in the single NAL unit mode, each NAL unit is framed in an RTP packet and they are transmitted in decoding order. The same transmission policy applies to the non-interleaved mode, but here NAL units pertaining to one access unit may be aggregated in single-time aggregation packets (STAP). This enables more efficient transmission (less overhead) for applications in which the size of NAL units is small on average compared to the MTU of the network. Further, NAL units may be fragmented; this is done if the size of a slice partition, data partition or parameter set is larger than the size of the MTU on the underlying network. If one fragmentation unit, such as a FU-A (fragmentation unit of type A) is lost, then the entire corresponding NAL unit is corrupted, and has to be discarded completely.

Lastly, the interleaved mode allows the transmission order of NAL units to be different from the decoding order. Since an H.264/AVC decoder assumes it is given NAL units in decoding order, a decoding order number (DON) has to be provided for each NAL unit. This packetization mode also supports aggregation of NAL units from different access units, so-called multi-time aggregation packets (MTAP). Obviously, the interleaved mode has the highest complexity and is not suitable for low delay applications.

In this work, only the non-interleaved packetization mode is considered, and no aggregation of NAL units is performed.

RTP payload format for H.264/MPEG-4 SVC

For backwards compatibility, the draft specification for the RTP payload format of SVC in [38] is based on that of H.264/AVC; RFC 3984. In [39], Wenger et al. present and discuss the current state of the draft, as it is not yet finalized. Aggregation and fragmentation of NAL units is done in the same way as earlier, but some design issues related to packetization, and new signalling of scalable layers in SDP had to be addressed. With regard to packetization, the SVC payload format draft mandates the use of either the non-interleaved or interleaved

mode. The reason for this is linked to aggregation; when the representation of a video sequence at a certain bit rate is divided into possibly many scalable layers, the average NAL unit size decreases significantly compared to a single-layer scenario. Both of these modes, as opposed to the single NAL unit mode, support NAL unit aggregation, and hence can accommodate more efficient delivery.

In the case of unicast streaming, a streaming server may well transmit an H.264/AVC base layer together with H.264/SVC enhancement layers in one RTP stream (and hence, within one RTP session). In this case, both the non-interleaved and the interleaved packetization modes are allowed. For multicast delivery, the SVC payload format draft recommends that the layers of a scalable bitstream are transported in different RTP streams to facilitate e.g. bit rate adaptation by intermediate application-aware nodes in the network. To allow such a node to distinguish the different layers from another, both session multiplexing and SSRC (Synchronization Source) multiplexing can be employed, each having their own strengths and weaknesses [39]. In session multiplexing, the scalable layers are transported in different RTP sessions. In the case of SSRC multiplexing, scalable layers are transported within the same RTP session, but differentiated using the SSRC identifier field in the RTP header. In any case, the interleaved packetization is to be used. Receivers are able to reassemble the NAL units from different scalable layers in the correct order by using Decoding Order Numbers (DONs) that span all RTP streams (and sessions).

Finally, to facilitate easy manipulation of aggregation packets, a new NAL unit type is defined that provide a table-of-contents view into such packets. The PACSI (Payload Content Scalability Information) NAL unit is identical to the 4 byte SVC NAL unit header, but the semantics of its fields is designed to reflect the combined importance of the NAL units inside an aggregated packet.

3rd Generation Partnership Project (3GPP)

The 3GPP Packet-switched Streaming Service (PPS) includes specifications and recommendations for IP-based streaming applications in 3G mobile networks [27]. Amongst other things, it decides which media codecs and control protocols for signalling, capability negotiation and media protection that can be used in a 3GPP compliant system. Some aspects of the specification are related to adaptive media delivery. RTP must be used for media transport, while media-specific signalling and reporting is performed using SDP and RTCP Extended Reports. In addition to loss statistics provided by Loss RLE reports (run-length encoded data describing which RTP packets are lost/received) defined in [34], an RTCP application specific report block is defined. This provides information related to the operation of the client receiver buffer, such as current playout delay, available buffer space, and next RTP sequence number (NSN) and application data unit

(ADU) to be decoded[3]. Mechanisms for transmission rate adaptation are not specified explicitly. However, the feedback reports should facilitate the design and implementation of rate-adaptive systems.

Notably, the PSS specification also includes optional mechanisms to signal quality-related metrics. The metrics defined in [27] are

- *Initial Buffering Duration:* time delay until playback starts.
- *Corruption Duration:* period during which the received media is corrupted.
- *Rebuffering Duration:* duration of interruption in continuous playback.
- *Successive Loss:* number of RTP packets lost in succession.
- *Frame Rate Deviation:* Difference between correct and actual frame rate.
- *Jitter Duration:* period during which the absolute error in playback time is larger than 100 ms.

2.2.3 Congestion Control for Multimedia Streams

As mentioned earlier, properly designed congestion control mechanisms are vital for the efficiency, stability and overall well-being of a packet-switched network. In the Internet today, congestion control is applied in an end-to-end fashion [40], and as TCP traffic is by far the most common traffic type, it is important – at least for service deployments over today's Internet – that video applications also behave reasonably and fairly when mixed with TCP flows [41]. Congestion control for streaming media applications is a well-studied area, and a great deal of work has been done in developing, analyzing and improving *TCP-Friendly* equation-based protocols in traditional wired [41–48] and wireless network scenarios [49–51].

However, there are still some open issues. In [42], Vieron and Guillemot address the rate smoothness and real-time requirements for video streaming, in the case of the standard TCP-Friendly Rate Control (TFRC) protocol [52]. However, the sending rate still fluctuates substantially, which is undesirable, and the scheme relies on packet loss to perform efficient congestion control. While video applications can be designed to have resilience against packet loss (see section 2.2.4), it is advantageous to minimize the error rate when the system is operating in steady state. Finally, several of the schemes that try to make the transmission rate of TFRC more smooth, e.g. [43], have drawbacks of being less responsive to congestion.

Fairness is an essential issue in the context of distributed congestion control mechanisms. While there are different notions of fairness and what exactly

[3]For H.264/AVC, an application data unit corresponds to a NAL unit.

constitutes a fair sharing of resources in a network [53, 54], two distinct types receive most attention in the literature; *max-min fairness* [55] and *proportional fairness* [56]. The latter tends to take total system efficiency into account, penalizing flows that use more of the overall system resources when allocating bandwidth on a congested link, while the former tends to equalize the resources allocated to each user [4].

In the original contribution by Kelly [56], a congestion control algorithm for achieving proportionally fair sharing was presented. However, it required explicit feedback from routers in the network. Later, Mo and Walrand [57] presented an end-to-end framework for obtaining such bandwidth sharing. Assuming a fluid model of all flows in the system, FIFO (first-in-first-out) queues and instantaneous feedback, a decoupled fairness criteria based on each flow's *backlog* – the number of packets pertaining to a given flow that occupy router buffer space along its path – was used to construct a end-to-end window-based proportionally fair congestion control algorithm.

Extending on this work, Xia et al. [58] – using the term *accumulation* to denote a flow's backlog on the forward path – describe a family of protocols that can be used to achieve proportional fairness as long as accumulation is used to control the flow rate. This will be presented in more detail next.

Accumulation-Based Congestion Control

The common property of accumulation-based protocols is that each flow should try to maintain a steady state accumulation inside the network. Such protocols can be described by the following differential equation [58], where $\dot{w}(t)$ denotes the change in congestion window.

$$\dot{w}(t) = -\eta \cdot g(t) \cdot f(a_i(t) - a^*) \tag{2.1}$$

In general, when the estimated accumulation $a_i(t)$ of a flow i is low, the sending rate is increased until some target accumulation a^* is reached, after which the sending rate is reduced to drain the excess accumulation. In Equation 2.1, η is a fixed positive rational number, $g(t)$ is some positive function, and $f(t)$ is a nondecreasing function with a single root for $f(0) = 0$.

Given sufficient buffer space in network routers, accumulation-based protocols can operate without packet loss in the steady state. Well-known issues with such schemes, beside the buffer size requirement, is that the estimate of accumulation on the forward path becomes unbiased when there is reverse path congestion,

[4] Note that when there is only a single bottleneck, both max-min and proportional fairness leads to a resource allocation in which all flows sharing the bottleneck link are given an equal amount of bandwidth

and the round-trip-time (RTT) cannot be estimated correctly. Also, due to the nature of the schemes, the round-trip propagation delay, as used in the control in [57], is inherently difficult to predict because buffer queues are non-empty in the steady state. The authors in [58] solve these problems by forwarding dedicated control packets through a high-priority queue in network routers and estimate the accumulation at the receiver side.

2.2.4 Error Control and Recovery for Multimedia Streams

Multimedia applications can be designed to be robust against packet loss, and this is a research area receiving a lot of attention [59]. Different strategies can be categorized based on whether they are host-centric, network-centric, or a combination. Further, among the host-centric schemes, some rely on adding a controlled amount of redundancy to combat loss at the sender side, while others are based on receiver-side error recovery.

Receiver-based schemes include retransmission of lost packets (e.g. delay-constrained automatic repeat request (ARQ) [60] and variants [61]), packet loss error concealment [29], and adaptive playout techniques [62]. The author has actively participated in research on error concealment; in [63], Bopardikar et al. presented a temporal error concealment technique that integrates the concept of *structural alignment* with traditional side-matching algorithms to better align edges in video content, thereby improving the performance both objectively and subjectively.

Among the sender-based schemes that add redundancy prior to transmission, some techniques employ this at the source coding stage, while others add redundancy to coded media data. Examples of the former include error resilience tools in video coding standards (for H.264/AVC, see section 2.1.1), multiple-description (MD) coding [64] and Wyner-Ziv coding [65]. Similarly, examples of the latter include forward error correction (FEC) and unequal error/loss protection (UEP/ULP). These schemes take advantage of the fact that media data may have different importance, which make them suitable tools in combination with MD [66] or scalable video coding frameworks [67–70].

A set of complementary techniques attempts to minimize the error at the receiver by deciding on the optimal packet scheduling. The first contributions considered a rate-distortion optimization framework [71–73], while congestion on a bottleneck link was additionally taken into account in [74] and [75].

While the schemes mentioned above are host-based in that they do not rely on any functionality or QoS support from the network, they may also be combined with prioritized transport such as service differentiation (DiffServ [76]), as in [77], or more complex schemes relying on network router feedback [78].

2.3 Towards User-Aware Visual Communications

In the previous sections, some of the components and technologies that enable highly efficient, adaptive and error-resilient visual communication systems have been presented. Today, the performance of different tools are often evaluated in simulation scenarios with varying level of realism. Often, the models do not reflect the complexity that characterize real-world systems. Further, in the design and evaluation process, more emphasis needs to be placed on how end-users experience the operation of the system, and the different trade-offs that have been made in designing it. Therefore, after realistic experiments have been performed, it is vital that the effects of compression and transmission can be presented to the end-viewer, e.g. in subjective tests involving human subjects. While Part A deals with video quality assessment and monitoring, Part B describes an experimental testbed infrastructure that can be used to facilitate such tests.

The evaluation procedure should result in quantifiable measures of end-user quality. These measures could then be used to configure and fine-tune the system, so that users get the best possible experience when viewing the rendered media content. Moreover, measures that can be used for real-time in-service monitoring of quality in deployed systems are particularly interesting. This can be exemplified in the case of the adaptive streaming solution presented in Part C. Rather than solely using a pre-decided adaptation policy (choosing which scalable layers to transmit) when adjusting the transmission rate – which is done in Part C – a perceptual no-reference quality metric could assist in deciding on the optimal adaptation policy on-the-fly. This could be helpful in improving the performance from an end-user point of view, together with being a valuable tool for quality assurance, which is in the interest of both end-users, content providers and service providers.

To this end, a significant amount of research on perceptual quality, video adaptation and end-to-end quality of service has been performed. Some references can be found in [47,59,79–86], However, many issues are yet to be addressed, and these topics will likely remain an active area of research for years to come. This thesis attempts to make some contributions within this area, specifically related to experimental ways of evaluating service performance, and providing end-users with a smooth and continuous streaming media experience even when the network distribution system is heavily loaded.

Part A

Video Quality Assessment

Part A

Video Quality Assessment

This part of this thesis deals with assessment of visual quality, and presents a method that can be used to estimate the video quality at the receiver side of a multimedia communication system. More specifically, the proposed metric estimates the severeness of block-edge artifacts in compressed video without using the original video as reference. Therefore, it may be integrated in a more sophisticated system for monitoring end-user perceived quality. First, an introduction to video quality and different assessment techniques are given.

A.1 Introduction

An increasing demand for ubiquitous access to multimedia content, and a corresponding increase in the variety and amount of content being produced, end-user terminals and networking facilities, calls for a solution which can facilitate a good user experience of media consumption. Some essential aspects of this problem are being addressed through the concept of Universal Multimedia Access (UMA) [3] which deals with the delivery of images, video, audio and multimedia content in general under various network access and resource conditions, communication device capabilities and end user preferences.

The objective of UMA enabled systems is to provide the user with the best possible subset of a multimedia resource that the user is capable of receiving. In this sense, UMA deals with quality with respect to the delivery of content. The quality is treated as an end-to-end Quality of Service aggregate which can be viewed as *Quality of Experience* (QoE). Increasingly, this idea is evolving to include the end-user and the user's perception of the media being delivered. In this premise, known as the Universal Multimedia Experience (UME) [5], the network and the terminal are considered purely as means to deliver the content.

The aim of this paradigm shift is to enable adaptation of the media content presented to the end-user based on that user's perception of that content in a specific environment and context. In other words, UME emphasizes the end user, and the ultimate goal is to provide him or her with meaningful content that maximizes the user's QoE.

When designing next-generation audiovisual communication systems, an essential design objective is therefore to maximize the end-users perceived quality of the provided service. However, to quantitatively measure an end-user's perception of an audiovisual service quality in an automated way is extremely difficult, and depends on factors that are not easily modeled. Human factors may include such diverse topics as user expectation and preferences, and psychophysical effects like perceptual organization. Similarly, factors related to the service, e.g. the physical environment and semantic context in which it is being used, will also play an important part in how the service is perceived. On the other hand, factors related to the representation and transmission of audiovisual signals are often more easily measured. However, such technical system parameters – at least yet – seem incapable of capturing all aspects of perceptual quality.

For these reasons, subjective testing is often considered the only viable option when it comes to evaluating audiovisual system performance and creating models for predicting end-user perceived quality.

A.1.1 Related Work

In order to quantify the perceived quality of a visual sensation through subjective testing, several factors need careful consideration. First, it is necessary to decide what to measure, which is known as the *attribute*. Further, one needs to hypothesize the cause for a (difference in) sensation, and finally, find an appropriate way to measure it. This leads to a choice of, or design of, a test *methodology*. Different methodologies are presented next, followed by a review of objective video quality measures and related studies.

Subjective Measures

For video, two main assessment methodologies exist, which differ in the way the test material is presented to participating subjects. When using the *single stimulus* method, test sequences are presented sequentially in a random fashion, either once or multiple times, and all test clips are rated independently. This method reflects a typical end-user scenario, where viewers do not have the possibility of comparing the viewed video sequence with a reference signal. In the *double stimulus* method, two test clips are presented to the viewer simultaneously, and the rating reflects a comparison between the two clips.

Further, for each of the two test methodologies above, both continuous quality evaluation (CQE) and post-presentation single quality evaluation (SQE) rating techniques can be employed. Yet another type of classification can be done based on the type of rating scale used in the test. For example, absolute category rating (ACR) is a popular choice for single stimulus testing, while the use of impairment scales (IS) are often more suitable for double stimulus tests. In ACR, subjects are asked to rate the video clip independently using e.g. the 5-point mean opinion score (MOS) scale, while for IS, they would rate the perceived impairment – or difference – between two video clips shown side-by-side.

These methods are described and standardized in a number of ITU recommendations, e.g. ITU-R BT.500-11, which deals with assessment of television picture quality [87], and ITU-T P.910 [88], which deals with video quality in multimedia applications. Further, ITU-T Recommendation P.911 [89] describes subjective assessment methods for evaluation of audiovisual quality in multimedia applications. The above recommendations typically outline characteristics of source sequences to be used as well as viewing and/or listening conditions.

Objective Measures for Video Quality

A generic UMA-enabled communication device used to consume a multimedia presentation needs to incorporate an awareness of UME, resulting in an intelligent behavior regarding how the content is presented, delivered, and ultimately, how the media is perceived by the end user. As previously noted, the latter is a subjective attribute that depends on several sensory factors that are not completely understood and are difficult to evaluate. Nonetheless, there is a clear need for automated evaluation of perceived quality of the rendered presentation. Specifically, a metric is required that will give us a quality measure that is strongly correlated with how the content is perceived by a cross section of end-users.

In general, such a metric would have to satisfy certain conditions. For one, the quality metric would have to have a low computational complexity. It would also be required to perform consistently over a wide range of content types. In many situations such as streaming of video, one would require a metric that could evaluate the perceptual quality of the content with either limited or no access to reference media. Such metrics are called *reduced-reference* (RR) and *no-reference* (NR) metrics, respectively [90, 91]. Metrics that estimate the perceived quality using the uncompressed original media as reference, are called *full-reference* (FR) metrics.

The most commonly used metric in image and video compression and transmission is the Peak-Signal-to-Noise Ratio (PSNR). It is a purely mathematical model involving the mean square error (MSE) between pixel values of two intensity images, and the maximum range of pixel values MAX_I. When PSNR is only calculated for the luminance (or luma) component of an image, it is denoted

Y-PSNR in this thesis. For a sequence of N video frames, in this work, the average sequence PSNR is calculated as follows;

$$PSNR_{seq} = \frac{1}{N}\sum_{n=1}^{N} 20 \times \log_{10}\left(\frac{MAX_I}{\sqrt{MSE_n}}\right)$$

Many attempts have been made to develop more suitable metrics for video quality. In most cases, properties of the human visual system (HVS) are utilized in order to design metrics that better reflect end-user perceived quality. An extensive coverage is provided by Winkler in [92]. Other important contributions include the NTIA (National Telecommunications and Information Administration) general model developed by Pinson and Wolf [93], the structural similarity index (SSIM) proposed by Wang et al. [94], the information-theoretic approach by Sheikh et al. [95],the Digital Video Quality (DVQ) metric proposed by Watson et al. [96], and the work of Masry et al. [97].

As mentioned above, NR metrics are useful in scenarios where access to the reference video stream is not available, such as in-service video quality monitoring. With no reference video signal to compare with, NR metrics often attempt to quantify the effects of various distortion artifacts. In particular, for block-based video compression schemes such as the MPEG and ITU standards (e.g. MPEG-1/2/4, H.263), the main forms of distortions include blocking effect, blurring, ringing and the DCT basis image effect [98, 99]). While sharpness metrics [100], and blur and ringing metrics [101] have been proposed, the main emphasis have been on quantifying the effects of blocking artifacts [91, 102–104]. This is because blocking artifacts tend to be perceptually the most significant of all coding artifacts [102].

The NR metrics described above are spatial-domain metrics, i.e. processing and feature extraction is done in the spatial domain on pixel values. Also, it is important to note that they predominantly consider compression-related distortions, and are not optimized for estimating the impact of transmission-related distortions. However, recent studies in [105] showed the capabilities of some NR metrics to accurately estimate the quality of JPEG2000-coded images subjected to bit-errors. The next section presents related work that also considers transmission-related distortion.

Objective Measures for Networked Media

Reibman et al. presented an approach to predict the SNR at the receiver side in case of packet loss [106]. Three different techniques were proposed, that differed in how much information is extracted from the received or incoming video bitstream. The frequency-domain method was shown to be quite effective, while the simpler methods relying on video header parsing, or solely on packet

loss, were not as efficient. However, flow-based measurement techniques remains an interesting research topic as they are light-weight, and scale well to higher-resolution video formats, and to a larger number of video streams if quality monitoring is performed in a centralized way (intermediate node in the network).

In another contribution [107], Patel et al. investigated effects of ATM networks impairments such as bit-errors and cell-loss on audiovisual quality of MPEG-1 and MPEG-2 video , and developed models of perceived audio, video and audiovisual quality based on regression analysis.

In [108], Vorren employed the ACR-HRR method to evaluate the perceived quality of H.264/AVC video subjected to packet loss in simulated Best Effort (BE) and Differentiated Services (DS) IP Networks [1]. The results indicated that service differentiation and the use of random early detection drop policies in the DS network only resulted in small gains in end-user perceived quality, and then only for rather high packet loss rates (5-10%). In addition, the study found that well-known objective metrics like the NTIA General Model [93], and even the peak-signal-to-noise ratio (PSNR), correlated quite well with the perceived quality of video with packet loss impairments (Pearson correlation around 90%). While the usefulness of PSNR is limited in a real-world delivery scenario, the NTIA General Model has the potential of being converted to an RR metric, and as such it may be useful in this context.

Finally, the Video Quality Experts Group (VQEG) within ITU is currently planning subjective tests in an attempt to validate and standardize objective video quality metrics for use in multimedia applications [109]. Other related work items of VQEG include a FR and NR metrics standardization effort for television applications. Previous studies include an evaluation of full-reference (FR) metrics for broadcast television monitoring systems [110], which was inconclusive in the sense that no single metric performed statistically significantly better than PSNR for all test cases.

A.1.2 Outline and Credit

This remainder of part A presents a simple method for estimating block-edge impairments in reconstructed video. A corresponding no-reference video quality metric is proposed in section A.2.1, while its performance is evaluated in section A.2.2.

The results presented next were published in [111] and [112]. The technique for detecting block-edge impairments was originally proposed by Dr. R. Venkatesh Babu, and was further developed and revised in close collaboration between the co-authors. The author and Dr. Ajit S. Bopardikar carried out performance

[1] The author of this thesis had the role of student advisor for this project.

simulations and improved the configuration of the proposed NR video quality metric.

A.2 No-reference Video Quality Estimation

Most algorithms that measure block-edge impairment – a.k.a. blockiness – make use of the fact that block-edge gradients can be masked because of spatial activity around them (spatial or texture masking), or may not be discernible in very dark or bright regions [91, 102, 104]. Block-edge gradients are typically computed as a function of the abrupt change in pixel values across a horizontal or vertical block-edge. Spatial activity is the degree of variation in pixel values in an area of the image, for instance the variation inside a block or near a block boundary. The higher the variation, the higher the spatial activity and better is its capacity to mask block-edge impairment. Thus, ideally, an NR blockiness metric should measure the users perception of blockiness in each video frame and do so with low computational complexity so that it can be used for real-time monitoring. In the next subsection, we describe a novel low-complexity blockiness metric based on the ideas mentioned above. At this point, note that block-edge impairments are less prominent in H.264/MPEG-4 AVC due to the in-loop de-blocking filter applied at the encoder and decoder side. As such, the method is more relevant to video encoded using e.g. MPEG-4 Part 2 Visual, MPEG-1/2, or H.261/3.

A.2.1 Proposed NR Block-edge Impairment Metric

As stated above, the metric proposed in this work is based on the idea that a block-edge gradient can be masked by a region of high spatial activity around it. It can be observed that blockiness perceived in a frame is usually because of blocks with at least one edge exhibiting low activity. Let B_{ij} represent an 8×8 block of pixels starting at location (i, j) in a given frame. I_k, $k = 1, \ldots, 4$, represents the edges of the block as shown in Figure A.1.

To measure the activity along a given edge I_k, the edge is first divided into three segments of length 6, namely, a_{k1}, a_{k2} and a_{k3}. This is shown in Figure A.2.

$$\begin{aligned} a_{k1} &= I_k(n) : n = 0 \ldots, 5 \\ a_{k2} &= I_k(n) : n = 1 \ldots, 6 \\ a_{k3} &= I_k(n) : n = 2 \ldots, 7 \end{aligned} \tag{A.1}$$

We define activity as the standard deviation, σ_{kl} for each a_{kl}, and $l = 1, \ldots, 3$. For a given edge I_k, activity is defined to be low if at least one of σ_{kl}, $l = 1, \ldots, 3$,

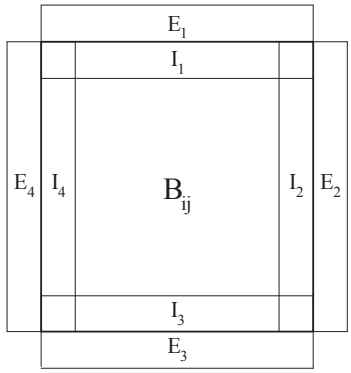


Figure A.1: An 8×8 block and its edges.

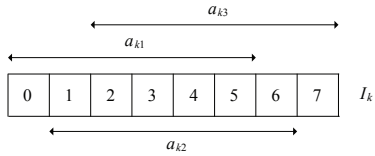


Figure A.2: A block edge of length 8.

is below a chosen threshold ε. In other words, if there is at least one segment of the edge, which has low activity (standard deviation), then the edge and thus the block it belongs to may contribute to the overall perception of blockiness in the frame.

The metric is then computed as follows. For each frame:

1. Initialize the block counter $C_B = 0$.

2. In each block B_{ij} along each edge I_k, for each a_{kl}, $k = 1, \ldots, 4$ and $l = 1, 2, 3$ compute the standard deviation, σ_{kl}. Thus we obtain three activity measures per edge giving us a total of twelve activity measures.

3. Now compute the gradient corresponding to each a_{kl}

$$\begin{aligned} \Delta_{k1} &= \text{mean}|I_k(n) - E_k(n)| : n = 0 \ldots, 5 \\ \Delta_{k2} &= \text{mean}|I_k(n) - E_k(n)| : n = 1 \ldots, 6 \\ \Delta_{k3} &= \text{mean}|I_k(n) - E_k(n)| : n = 2 \ldots, 7 \end{aligned} \tag{A.2}$$

where E_k, $k = 1, \ldots, 4$ are the edges adjacent to the corresponding block

edges I_k, as shown in Figure A.1.

4. If at least one segment satisfies

$$\begin{aligned} \sigma_{kl} &< \varepsilon \\ \Delta_{kl} &> \tau \end{aligned} \tag{A.3}$$

$k = 1, \ldots, 4$ and $l = 1, \ldots, 3$, increment C_B by 1. That is, we count B_{ij} as contributing towards the overall perception of blockiness of the frame.

The overall blockiness measure $\mathcal{B}_F$ for the present frame, is then

$$\mathcal{B}_F = \frac{C_B}{\text{Total number of blocks in the frame}}. \tag{A.4}$$

Clearly, the range of the metric is $[0, 1]$ where a value of 0 corresponds to no visible block-edge impairment, and increasing values of B_F implies increasing block edge impairments in reconstructed video frames.

The bit depth for the video sequence is assumed to be 8 bits or 255 gray scale levels. The value of ε is chosen as a threshold to isolate edges with low activity. To this end, a value of $\varepsilon = 0.1$ is chosen. This corresponds to the situation when there is a minimal deviation from the mean of the segment. Increasing the value of ε would result in edges with a greater standard deviation being picked. This would mean picking blocks with segments that might have enough spatial activity to mask the block-edge gradient for that edge.

The value of τ can be chosen so that given low activity, the largest number of perceivable block-impaired edges will be counted in the metric. Increasing the value of τ would mean rejecting segments with low spatial activity which also have a block-edge gradient that can be perceived. On the other hand, choosing a very small value of τ would result in a situation where an imperceptible edge might result in a block being counted, thus giving a false reading. For the simulations presented next, a value of $\tau = 2.0$ was used. This specific value of τ was observed to give the best performance for a wide range of video sequences.

A.2.2 Performance of the Proposed Metric

In the simulations, 10 second video sequences in CIF resolution (frame size of 352×288), 30 frames/sec and YUV (4:2:0) format, was used. For results presented here, only the Y or the luminance channel is used in the algorithm. The original video sequence was encoded at various bitrates using the XviD MPEG-4 ASP encoder [113] with a GOP size of 30 frames.

The performance of the proposed metric is compared with the Wang, Sheik

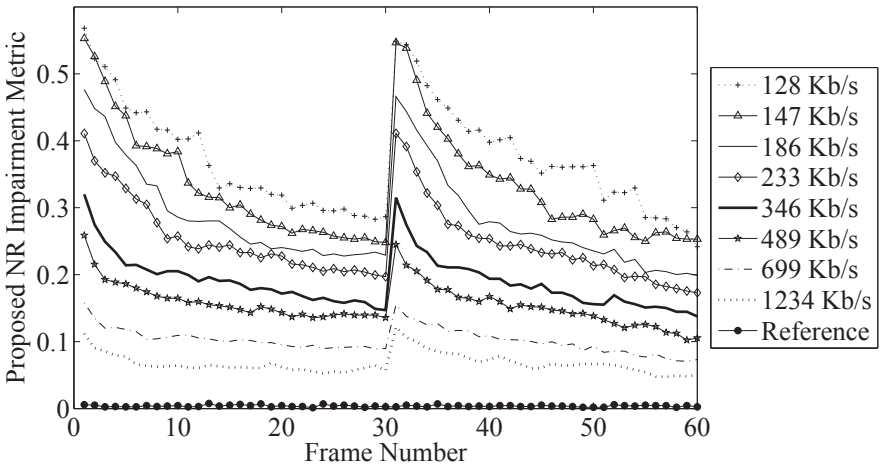


Figure A.3: Proposed blockiness metric for the first 60 frames of the "Paris" sequence coded at different bitrates.

and Bovik (WSB) quality assessment model [2] [103]. Because the WSB metric increases with image quality, and typically has range of 0 to 10, the WSB model output is normalized by 10 and this result is then subtracted from 1. This procedure enables an intuitive comparison with the proposed metric.

Both metrics were computed for each frame of the original and the encoded sequences. Here, results obtained for the "Mother-Daughter", "Paris" and "Stefan" sequences are presented. Figure A.3 shows the result of applying the proposed NR metric to the first two GOPs (frames 1-60) of the "Paris" sequence and Figure A.4 shows the corresponding results for the WSB metric. Note that the proposed metric is nearly zero for the original sequence. In other words, it measures no block-edge impairment in the uncompressed original video as expected. At the same time, the measured distortion increases as compression increases, or equivalently, the bitrate decreases.

It can also be observed that both metrics increase as the compression increases – or equivalently – the bitrate decreases. This is in keeping with the fact that higher compression implies coarser quantization and consequently increased perceived blockiness. The peaks in both figures indicate the I (intra-coded) frame. The peak suggests that blockiness perceived in the I-frame is the highest in a GOP

[2] Matlab code for the model was obtained from Zhou Wang's website at "http://www.cns.nyu.edu/∼zwang/"

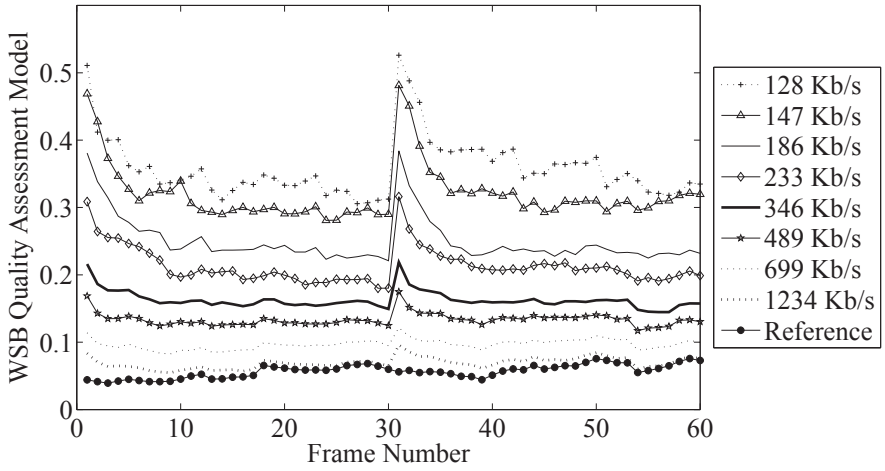


Figure A.4: WSB metric for the first 60 frames of the "Paris" sequence coded at different bitrates.

at all bit rates.

Figure A.5 shows the change in both metrics for one frame, namely, frame number 31 which is an I (intra-coded) frame encoded at different rates, namely, 1234 Mbps, 699 kbps, 489 kbps, 346 kbps, 233 kbps, 186 kbps 147 kbps and 128 kbps. It can be seen that both curves show a graceful behavior, and that the measured block-edge impairment decreases with increasing bitrate, as expected. Note that the WSB model output is transformed to an impairment metric for this plot, as discussed above. Particularly, the maximum WSB value of 10 is not universally applicable, but was considered suitable for this sequence. Hence, the discrepancy between the two graphs does not indicate a significant difference in performance, and attention should rather be given to the slope of the curves.

While evaluating the performance of the proposed metric for different types of video content, it became evident that the metric performs the best for sequences with low degree of camera movement and small amount of object motion.

Figure A.6 shows the performance of the proposed metric for the first 60 frames of the "Mother-Daughter" sequence. This sequence depicts typical head and shoulder type of video content with a fixed positioned camera. Again, note that the metric is nearly zero for the original uncompressed video frame. Also, note that the metric attains its maximum for frame number 31 which is the I-frame.

Similarly, figure A.7 shows how the metric perform for the "Stefan" clip,

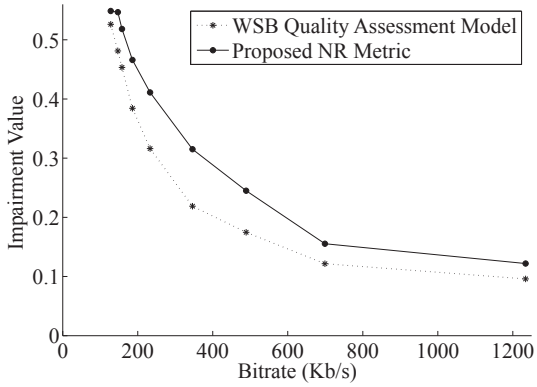


Figure A.5: Comparison of the proposed metric and the WSB metric for frame 31 of the "Paris" sequence at different bit-rates.

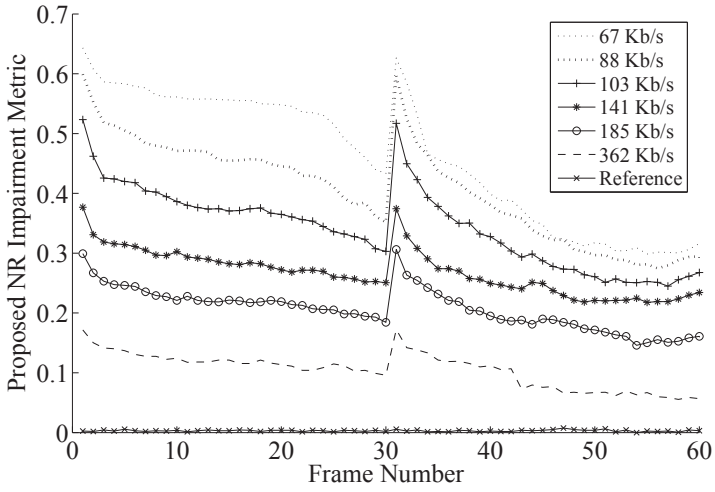


Figure A.6: Proposed metric for the first 60 frames of the "Mother and Daughter" sequence coded at different bitrates.

which has a significant amount of camera motion, and complex object motion. As can be seen, the proposed metric still measures a considerable amount of block-edge impairments in intra-coded frames. However, for the predictively encoded frames, little block-edge impairment is detected. From visual inspection, it was

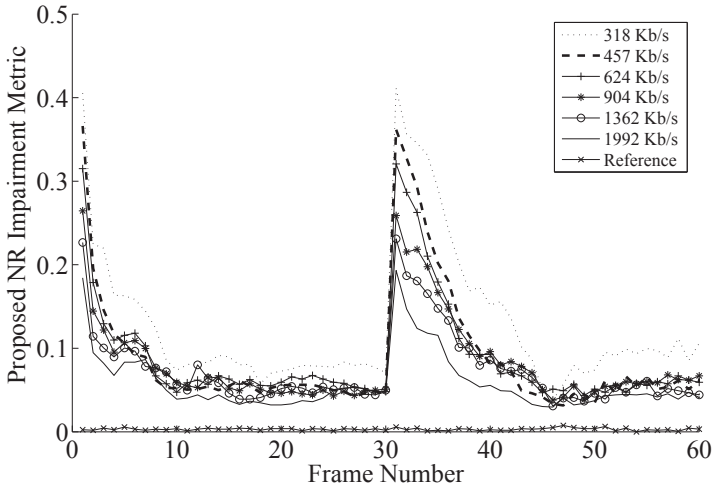


Figure A.7: Proposed metric for the first 60 frames of the "Stefan" sequence coded at different bitrates.

verified that high-motion sequences like "Stefan" have smaller amount of block-edge impairments than low-motion sequences. The reason for this is that the motion compensation process has a more significant effect on the location of the block-edge impairments. Although the reconstructed residual in P-frames may contain blockiness, and the previous I/P-frame from which it is predicted contains blockiness, the motion compensation will tend to move and smear these impairments. Because of this, the impairments will to a less extent be aligned with the edges of 8 by 8 blocks, and additionally, due to the smearing effect, they will also be less prominent closer to the end of a GOP. Note that this does not imply that P-frames have higher visual quality than I-frames, but simply that block-edge impairments are weaker and less prominent, and explains why the metric does not detect the same amount of blockiness in these frames.

Figure A.8, shows frame 31 (I-frame), frame 40 and frame 55 of the "Mother-Daughter" sequence, encoded at 88.5 Kb/s. As the blockiness metric decreases from 0.62 for frame 31 in figure A.9 to 0.28 for figure A.8(b), the perceived blockiness in these frames also decreases. In addition, note that other impairments such as blurriness and ringing start to play a more dominant part in the overall perception of the frame.

Figure A.9 shows one frame, namely, frame number 31 which is an I (intra-coded) frame encoded at three different rates, namely, 362 Kb/s, 141 Kb/s and 89 Kb/s, along with the original. The corresponding blockiness metrics are given

(a) Frame 31 (I) ($\mathcal{B}_F = 0.62$)

(b) Frame 40 (P) ($\mathcal{B}_F = 0.38$)

(c) Frame 55 (P) ($\mathcal{B}_F = 0.28$)

Figure A.8: Three sample frames from the "Mother-Daughter" clip.

in the caption to the figure. One can see substantial blockiness in the Figure A.9(d). The corresponding value of the blockiness metric here is 0.62. Likewise, as the perceived blockiness decreases from Figure A.9(c) to Figure A.9(a) the blockiness metric decreases from 0.37 to 0.001 for the original.

As mentioned above, the metric performs particularly well for I-frames, in which block-edge impairments constitute the dominant compression-related distortion. To show the capabilities of the metric to estimate the received quality of intra-coded frames for different types of video content, its performance is compared to the Structural Similarity Index (SSIM) proposed by Wang et al. [94]. SSIM is a full-reference quality metric that has a range between 0 and 1, where a value of 1 corresponds to the best picture quality. Figure A.10 compares the performance of the proposed quality metric – which is simply $(1 - \mathcal{B}_F)$ – to SSIM and PSNR for frame 31 of the "Paris" sequence. Figure A.11 shows a similar comparison for frame 181 of the "Stefan" sequence. Since the output values of the three models differ, attention should first of all be given to the

(a) Frame 31 original ($\mathcal{B}_F = 0.001$) (b) Frame 31 (I) 362 Kb/s ($\mathcal{B}_F = 0.17$)

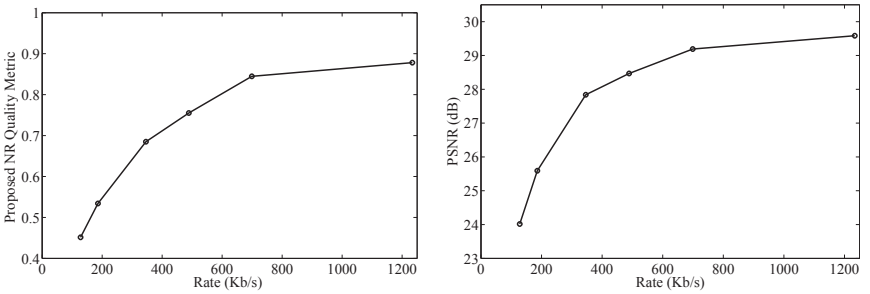
(c) Frame 31 (I) 141 Kb/s ($\mathcal{B}_F = 0.37$) (d) Frame 31 (I) 89 Kb/s ($\mathcal{B}_F = 0.62$)

Figure A.9: Frame 31 of the "Mother-Daughter" clip encoded at different bitrates.

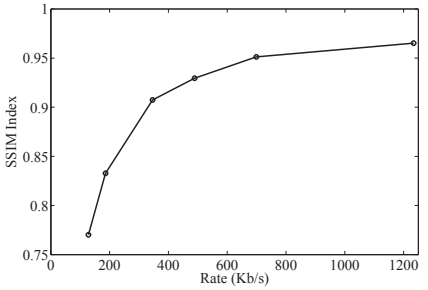
slope of the curves. The plots of Figure A.11 indicate that the curve for the proposed metric is somewhat less steep than the corresponding plot for SSIM and PSNR. This particularly applies for the "Stefan" clip, and indicate a less favorable performance of the proposed metric. However, in the case of the "Paris" clip in Figure A.10, the slope of the curves are more congruent.

A.3 Summary and Discussion

In this part, a No-Reference metric was proposed that measures block-edge impairments in reconstructed video. Examples of its performance have been presented. The results indicated that the proposed NR metric performs well for intra-coded video frames, in which block-edge impairments are the most dominant artifact, and as such, has the biggest and most severe impact on perceived quality. The metric can be used on the receiver side of a video communication system, and thus, it can be applied as an integral part of a real-time video quality monitoring



(a) Proposed NR Quality Metric (b) PSNR

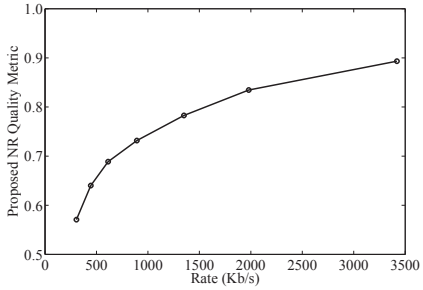


(c) Structural Similarity Index (SSIM)

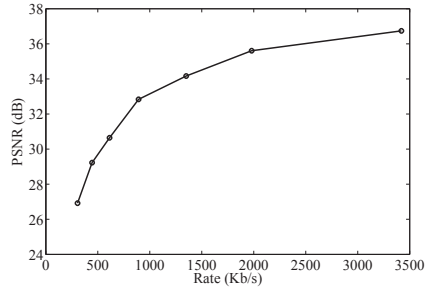
Figure A.10: Comparison of video quality metrics for the intra-coded frame 31 of the "Paris" sequence.

tool. In addition, the proposed method could also be used in other applications such as post processing of video frames for improved perceptual quality (e.g. error concealment, de-blocking filters).

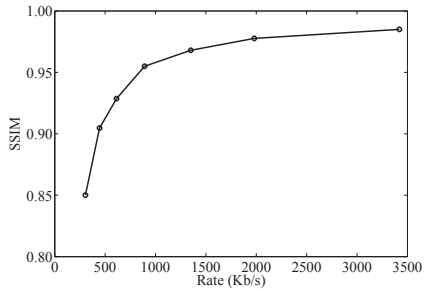
However, the approach also has some rather significant limitations. While the block-edge impairment metric performs well for sequences with low amount of camera and object motion, it does not properly capture the main distortions in predictively encoded P-frames for sequences with high motion. In these frames, block-edge impairments are less prominent. Further, with the standardization and adoption of H.264/MPEG-4 – which includes a mandatory in-loop de-blocking filter to combat such distortions – the proposed method may be less relevant in future systems. Still, the development of NR video quality metrics that estimate the impact of compression and transmission-related visual artifacts, remain an important research goal, and will be essential in developing fully automated quality monitoring and assessment systems. While comparison with state-of-the-art FR video quality models is an important and flexible way of



(a) Proposed NR Quality Metric



(b) PSNR



(c) Structural Similarity Index (SSIM)

Figure A.11: Comparison of video quality metrics for the intra-coded frame 181 of the "Stefan" sequence.

determining the performance of proposed NR metrics, they should also be validated using formal subjective assessment methodologies.

Part B

Streaming Media Testbed

Part B

Streaming Media Testbed

While part A considered how the visual quality of a deployed video service could be predicted at the receiver side through the use of no-reference video quality metrics, part B deals with performance evaluation of such streaming media services in controlled laboratory-type experiments. Issues related to the level of control, repeatability and measurements are discussed, and the proposed multimedia testbed is used to evaluate the performance of an IP-based H.264/AVC broadcast video service subject to packet loss.

B.1 Introduction

There are three main approaches for evaluating the performance of networked multimedia applications. First, the most common way is to build an accurate *simulation model* of the application's behavior, together with an appropriate network model, and evaluate the performance in a discrete event simulator like ns-2 [114]. The two other methods allow using real applications and end-systems – and are thus experimental in nature – but differ with respect to how realistic the attributes of the underlying network are. A *network emulator* typically provides a high-level abstraction of typical network behavior, while a test setup using *real network devices* provides the most realistic scenario. Disadvantages of using real applications and network devices are scalability issues when studying a larger number of users and/or network nodes, and a lower level of repeatability than what can be offered by network simulations.

B.1.1 Related Work

Most of the published work involving experimental testing of multimedia applications relies either on network emulation or measurements performed on the public Internet. In the case of network emulation, some basic assumptions about link bandwidth, network delay, delay jitter and packet loss usually have to be made. The idea of considering the entire network as a black box, in which certain high-level network characteristics are configured, is referred to as wide-area-network (WAN) emulation. While some network emulators only support WAN parameterizations, others allow more detailed configuration of buffer queue sizes and selection of different drop policies. Emulab, originally developed at the University of Utah, is a set of large-scale network emulation facilities available for research and testing [115], [116]. More often though, smaller-scale local experimental setups using e.g. the NISTnet emulator are used, as in [117] and [118], or some commercial alternative, as exemplified by the study in [119].

An approach for doing real-time emulation of wireless channels was presented by Kellerer et al. [120], in which packets were intercepted and redirected to the network model emulation software through the use of Linux Divert Sockets. Experiments involving network emulators provide more realism than simulation models, but not the same level of control and repeatability.

Recent studies within the European NEWCOM project consider increasing the level of repeatability in network emulation of wireless networks. Towards this end, a system is designed in which simulations are first performed offline, and later, the outcome of the simulation is used by the emulator in deciding which packets should be delayed by how much, and which are to be dropped [121, 122].

In the case of Internet experiments, the results obtained can be quite realistic and representative of an actual deployment. Viéron and Guillemot [42] used such experiments to verify the performance of a real-time extension to TFRC, and Wenger employed packet loss traces from Internet experiments to evaluate the performance of error resilience tools in the H.264/AVC standard [20].

However, there are some important issues with Internet experiments, as discussed by Spring et al. [123] in the case of PlanetLab [124] Firstly, results from such experiments are not easily reproducible, since the traffic on the Internet changes significantly on multiple time scales [125, 126]. If the cross-traffic experienced by a flow changes significantly from milliseconds and seconds, to hours, days and months, it is very hard to say exactly *when* it is appropriate and representative to perform experiments.

The second aspect is related to *where* the experiments are made; it is equally hard to make a test setup that naturally reflects the Internet in a representative way. This is due to its heterogeneous nature, and the fact that most of the nodes in wide-area experimental testbeds like PlanetLab are connected through high-

bandwidth research networks, a situation which does not necessarily resemble the dominant commercial networks. Note that Pucha et al. show how such distributed testbeds can be made more realistic by ensuring that traffic flows traverse both research and commercial networks [127].

Finally, from a social context, people's usage patterns change over time as new services such as peer-to-peer (P2P) file sharing and – more recently – user-driven video portals become more popular. Therefore, although Internet experiments are the most realistic way to evaluate the performance of multimedia applications, unless great care is taken, the presumptions on which the experiments are based may often be less valid. Therefore, in this work the focus is on increasing the level of realism and repeatability of controlled testbed experiments.

B.1.2 Outline and Credit

In this section, we present an IP-based streaming media testbed that can be used for evaluating the performance of streaming media applications, and to evaluate different network QoS strategies for such applications. The proposed testbed is flexible in that it consists of a set of off-the-shelf and open-source software components that can be put together to simulate or emulate a specific streaming scenario. Besides streaming servers and streaming media client software, components include a controlled test network, a hardware IP network emulator, a high-precision packet capture device and a packet flow regenerator to enable accurate recreation of packet flows and specific network conditions.

To be able to run repeatable tests and measurements in a controlled environment using real streaming media systems and real network devices, we need to verify the performance and precision of the testbed components. For instance, to be able to recreate specific network scenarios for interactive applications that require low bounds on delay (and buffer sizes), we must verify that the variable delay introduced by network emulation and packet stream regeneration will not considerably affect our results.

The remainder of Part B is organized as follows; Section B.2 describes the different network and end-system components of the testbed, while section B.3 gives some preliminary results of the performance of the packet regenerator. Section B.4 describes how the testbed is used to perform an error robustness evaluation of H.264/AVC video, and finally, section B.5 provides a summary and related discussion.

The results presented in the following have been published in [128], [129] and [130]. The streaming media testbed consists of a set of open-source and commercially available components, and has been designed and assembled in a coordinated fashion by a number of people at the Centre for Quantifiable Quality of Service in Communication Systems (Q2S, NTNU) and Uninett [131]. As

such, the contribution of the author concerned evaluating the performance of the packet regenerator device [128]. This work was carried out in close collaboration with Bjørnar Libæk, who performed software installation and patching of the Linux 2.6 Kernel. Finally, the error robustness evaluation of H.264/AVC – including development of the pcap2avc software – was carried out by the author, in consultation with the co-authors of [130].

B.2 Testbed Overview

Conceptually, the testbed consists of an *application part* and a *network part*, where the former is media aware, and the latter can be either media aware or unaware. Before describing each component, an overview of the network part is given.

In general, the network part of the testbed setup can consist of several *manipulation elements* $\{m_i\}_{i=0}^{n}$ which distort the original flow by introducing packet loss and delay. Examples are real routers and switches fed with competing traffic, software routers, network simulators, or network emulators. The elements will be chained together like shown in figure B.1.

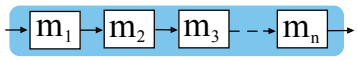


Figure B.1: Chain of n flow manipulation elements

The transfer of flows between elements can either be done online, by having a network link between the two elements, or offline, by capturing the flow and writing it to a *trace file* together with timing information for later manipulation. Further, one can say that each offline transfer divides the chain into two *phases*, which are separate in time. Typically, an offline transfer is necessary when the element on the right hand side is a Discrete Event Simulator, which because of its own time-scale needs to reside in a separate phase. After offline manipulation, it may be necessary to regenerate the packet flow using a *flow regenerator*. Regeneration is another phase separation point. It may be used to replay a captured flow into a router, or to replay an already manipulated flow for reception at the media decoding host.

Figure B.2 depicts one specific configuration of the testbed setup, which is used later in section B.4. The packet flow originates at the streaming server, and ends up at the decoding host. In the middle, there is a simple network part consisting of a single manipulation element, namely a network emulator. After passing through the emulator (phase 1), the flow is captured by a capture device, and later replayed by a flow regenerator (phase 2). None of the components in

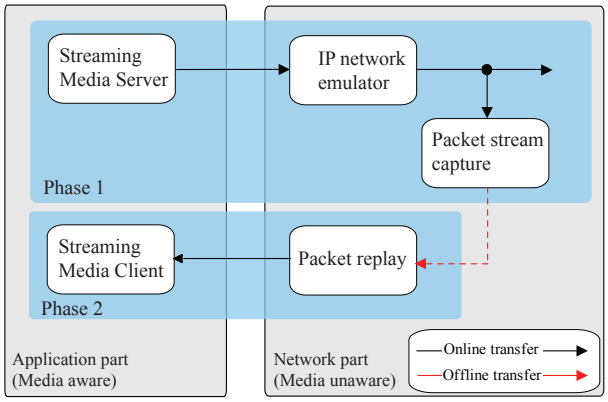


Figure B.2: Testbed overview

the network part are media-aware, i.e. all operations are performed on IP level or below.

One important objective of this testbed is to study the effect of packet loss and delay on packet flows containing media data in a controlled environment. It is therefore crucial that no unexpected non-measurable delay is introduced by any of the components. In section B.3.1, results are presented indicating the delay introduced by the packet regenerator. The rest of this section describes the components of the testbed in more detail.

B.2.1 Streaming Media Server

Streaming servers usually have both on-demand and broadcast functionality. As mentioned previously, for on-demand streaming, RTSP is the standard protocol for setting up and managing new sessions. In the case of broadcast streaming using a multicast IP service, SDP is often used for description so that new users can easily access the media being broadcasted. The media data packets are transmitted using the RTP protocol [132].

In this work, several RTP-based streaming servers and broadcast applications have been used. First, the work leading to a performance evaluation of the packet regenerator made use of the broadcast features in the Envivio 4Sight MPEG-4 Streaming Server (4-Sight) [133]. In this setup, the server transmitted pre-encoded MPEG-4 video over RTP/UDP [31] by parsing the hint track available in an MP4 file [134]. The framing and timing information available in this hint track decides how video data is mapped into RTP packets and sent across the test network. As the broadcast functionality in the streaming server was used,

no RTSP setup and negotiation was performed prior to transmission. Darwin Streaming Server [135] – an open-source variant of the Quicktime Streaming Server from Apple – has similar features, and could have been used as an alternative to the server from Envivio.

B.2.2 IP Network Emulator

Network emulation is a way of synthetically subjecting applications and hosts to real-world network impairments. As mentioned in section B.1.1, while network emulators can not match the controllable and repeatable characteristics of discrete event simulators, they offer real-time operation, are easily modifiable and offer better repeatability than measurements in live networks [116]. Network impairment patterns such as packet loss, delay, jitter and bandwidth constraints are some of the high-level parameters that can be configured. In this testbed, the PacketSphere Network Emulator from Empirix [136] was used.

One part of the user-interface for this emulator is shown in figure B.3. As can be seen, the emulator is configured to introduce 3 % randomly distributed packet loss, and a constant delay of 50 ms. In addition, the link bandwidth is restricted to 512 Kb/s. While the PacketSphere is a commercial hardware solution, freely available alternatives exist, such as the NISTnet open-source network emulator for Linux [137].

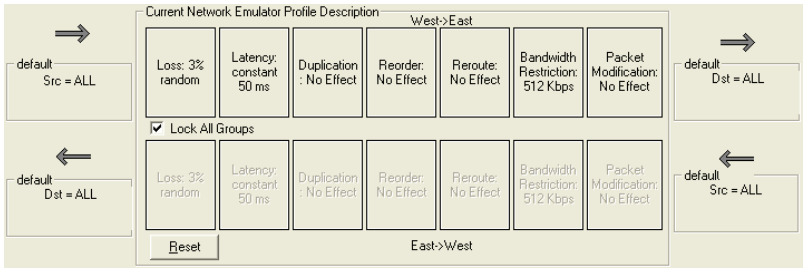


Figure B.3: IP Network Emulator GUI

B.2.3 Media Flow Monitoring and Capture

In order to perform accurate network link monitoring, delay measurements or traffic characterization, specialized *network monitoring cards* are often needed. They provide packet processing and capture functionality even on high-speed Gigabit links, and the timing resolution is several orders of magnitude better than what can be obtained on a desktop computer. In this work, the DAG 3.5E

network interface monitoring card from Endace was used [138]. These cards have several very useful capabilities; first, every packet and their entire payload can be captured and written to a trace file. Second, a microsecond precision time stamp reflecting arrival time on the link is attached to all packets. Note that the DAG cards can also be synchronized with GPS (Global Positioning System) signals. Several trace file formats exist; ERF (Endace Extensible Record Format) is the native format used in the DAG cards, but they also support the commonly used PCAP format [139]. The software package *dagtools* provides capturing and conversion tools, and neither data nor timing resolution is lost when converting between the two formats.

B.2.4 Media Flow Regeneration

A packet flow regenerator consecutively reads packets and their corresponding timestamps from a trace file and sends them out on a network interface. In this process, there are several possible sources of unwanted delay, depending on the operating system, hardware resources, system load and the implementation of the regenerator. Examples are disc access, timer resolution and the process scheduler in the operating system. To minimize these effects, one could use a real-time operating system with higher timer resolution and the ability to prioritize real-time processes. On the other hand, the delays introduced may not be significant for the test results. For instance, if the delays are much smaller than the decoder's playout buffer, they can most likely be ignored.

With this in mind, one of the purposes of this work was to study the performance of a flow regeneration tool called *tcpreplay* [140] running on a regular Linux operating system. Tcpreplay takes a PCAP trace file as input. Because of the real-time extensions of the 2.6 kernel, both 2.4 and 2.6 kernels were evaluated at different loads. The 2.4 kernel is a pre-built Debian sarge kernel image of version 2.4.26. The 2.6 kernel used was version 2.6.8 with the "Preemptible Kernel" option (CONFIG_PREEMPT=y) set. This option allows low-priority processes running in kernel mode to be interrupted by time critical events [141]. The following hardware configuration was used: 1 GB RAM, 3.0 GHz Pentium 4 CPU and SATA 7200 rpm 8 MB cache disks.

B.2.5 Real-time Streaming Media Client

As shown previously in figure 2.5, a streaming media client receives, reorganizes and buffers media data in a playout buffer. The client decoder fetches media data from the playout buffers, performs error detection, decoding and error concealment, so that the video frames or audio samples are available for rendering by a display/sound device at presentation time. Several popular and feature-rich open-source media players exist, e.g. the VLC player from Videolan [142]. It uses

the 3rd. party libraries *livedotcom* [143] and *libavcodec* [144] for RTP streaming and media decoding, respectively.

B.2.6 Offline Media Receiver - pcap2avc

In some scenarios, it is necessary to perform media decoding in a separate phase, after a network simulator has introduced impairments. This may also be advantageous in network emulation experiments if for instance no streaming clients with real-time decoding capabilities are available. This was indeed the case for the study presented in section B.4, in which an error robustness evaluation of high-definition 720p H.264/AVC video was carried out using the JM (Joint Model) reference software decoder from JVT [145]. For this work, an offline RTP receiver named *pcap2avc* was developed.

The structure of pcap2avc is depicted in Figure B.4, which operates as follows: packets are sequentially read from a PCAP trace file, and MAPI [146] is used to filter out the video packet flow. After parsing the MAC, IP, UDP and RTP headers, NAL Units (NALU) are assembled according to RFC 3984 [33]. NALUs that are correctly received are then written to an H.264/AVC Annex B byte stream file. As previously mentioned in chapter 2.2.2, corrupted NAL units have to be discarded.

The important parameter set NAL units are often transported offline using SDP, and as such, they would not be available in the PCAP trace file. Therefore, the description is input to the software together with the PCAP file. The SDP description can easily be extracted from the hint track of the source MP4 file using tools available in MPEG4IP [147] or Darwin Streaming Server [135] [1]. Finally, decoding of the .264 file can then later be done using the JVT JM reference software.

[1] As an implementation note, it is worth mentioning that the SPS and PPS parameter set NAL units in the SDP file are represented in binary form using base64 encoding, and that the PCAP format employs big-endian byte order, similar to how packets appear on the network link.

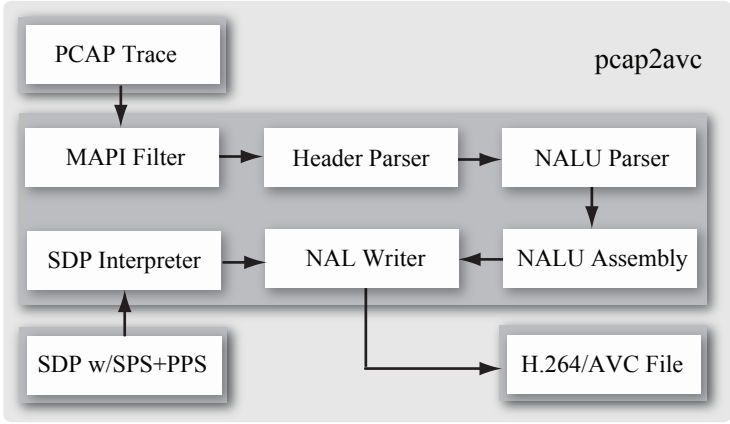


Figure B.4: pcap2avc Offline H.264/AVC RTP receiver

B.3 Testbed Performance Measurements

This section will present results from the process of verifying the performance of the individual testbed components. First, the performance of the packet flow regenerator is investigated. Some of these results were published in [128].

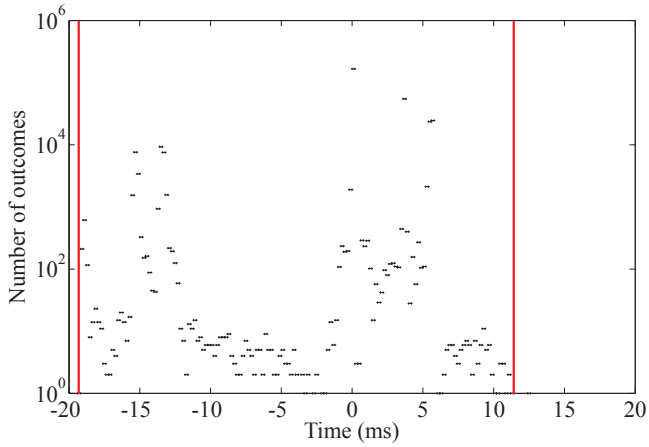
B.3.1 Performance of the Packet Flow Regenerator

In this section, two possible performance enhancements of the packet flow regenerator are investigated. The first is related to preemption in the Linux 2.6 task scheduler, while the other concerns an alternative packet transmission method introduced in a new software release of tcpreplay (on which the packet regenerator is based).

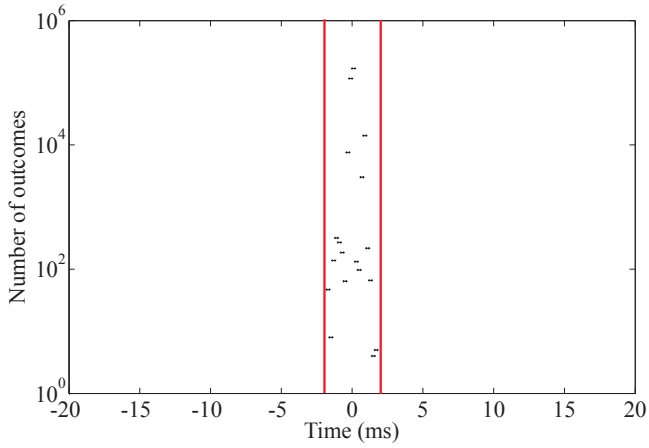
As mentioned in B.2.4, configuring the Linux 2.6 kernel with the "Preemptible Kernel" option set allows low-priority processes running in kernel mode to be interrupted by time critical events. Therefore, if tcpreplay is executed at the highest priority, kernel preemption would allow tcpreplay to interrupt lower priority tasks, and be scheduled immediately.

For these measurements, the entire StEM sequence downsampled to a resolution of 1024x416 pixels was used. The clip was encoded using Envivio 4Coder 3.0 [133] to MPEG-4 Advanced Simple Profile at various constant bit rates (CBR), with a intra period of 1 second (GOP size = 24) and encoder video buffer size of 1 second.

tcpreplay on Linux 2.4 vs Linux 2.6 kernel



(a) tcpreplay running on Linux 2.4



(b) tcpreplay running on Linux 2.6

Figure B.5: Distribution of difference in packet inter-arrival times comparing 5 Mb/s traces from 4-Sight and tcpreplay (Bin width equal to 0.5 ms for both traces)

To measure how accurately tcpreplay is able to regenerate packet flows, we compared a 5 Mb/s MPEG-4 network trace from 4-Sight with several network traces from tcpreplay regenerating the 4-Sight trace. Similar measurements were

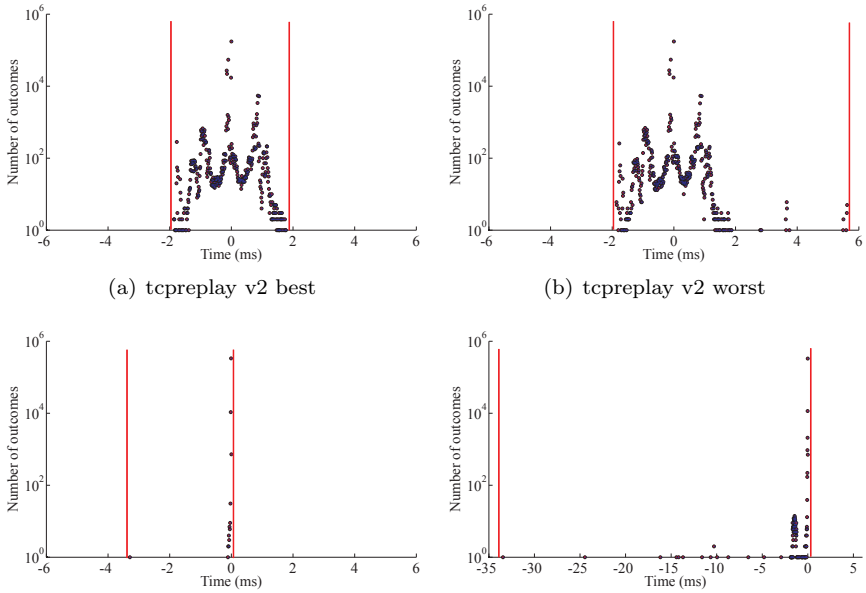
performed with tcpreplay running on both the Linux 2.4 and 2.6 kernel. These experiments, published in [128], were performed using version tcpreplay version 2.3.3.

Figure B.5(a) shows the distribution of differences in packet inter-arrival times between the traces from 4-Sight and tcpreplay running on the 2.4 kernel. Figure B.5(b) shows the corresponding distribution with tcpreplay running on the 2.6 kernel. As these plots clearly indicate, there is a considerable performance gain when using the Linux 2.6 kernel. The reasons behind this improvement are discussed in more detail by Von Hagen in [148]. In fact, considering the extreme values obtained from our measurements, tcpreplay running on a Linux 2.6 kernel is able to recreate the packet flow ten times more accurately than tcpreplay running on the 2.4 kernel. While tcpreplay seems unable to recreate the trace with a higher precision than ±20 ms on the 2.4 kernel, these bounds are around ±2 ms on the 2.6 kernel.

	Absolute difference in inter-arrival times (ms)		
Bit rate	Mean	99^{th} Percentile	Max
1 Mbps	0.261	0.953	2.10
5 Mbps	0.139	1.100	2.70
10 Mbps	0.077	0.914	1.80
15 Mbps	0.078	1.100	5.90
20 Mbps	0.073	1.200	24.5

Table B.1: Absolute difference in packet inter-arrival times between traces from 4-Sight and tcpreplay running on Linux 2.6 kernel, for different bit rates.

Table B.1 shows statistical characteristics of the absolute differences in packet inter-arrival times between traces from 4-Sight and tcpreplay running on Linux 2.6 kernel, for five different MPEG-4 flows encoded at 1, 5, 10, 15 and 20 Mb/s. We see that, e.g. for the 15 Mb/s flow, 99 % of packets in the regenerated packet flow are sent within ±1.1 ms of the intended sending time. The maximum error increases for higher sending rates, e.g. the worst-case difference between actual and intended sending time for the 20 Mb/s packet flow is 24.5 ms. It should be noted however, that only 3 out of 1244686 packets were sent more than 3.3 ms later than intended during this session. This phenomenon could be related to higher priority system processes forcing tcpreplay to sleep longer than intended [140]. Also, we see that the mean error in sending time decreases as the bit rate increases. For higher bit rates and an increasing number of packets per video frame, more and more packets are sent immediately succeeding each other, without tcpreplay having to sleep until the next calculated sending time. This way, the relative frequency of idle periods is reduced, and the rate of times at which tcpreplay has to wake up at the exact right moment is reduced.



(a) tcpreplay v2 best

(b) tcpreplay v2 worst

(c) tcpreplay v3 best

(d) tcpreplay v3 worst

Figure B.6: Distribution of difference in packet inter-arrival times comparing 5 Mb/s traces from tcpreplay v2 and v3 (Bin width equal to 0.01 ms)

Performance of tcpreplay version 3

Since the initial study of the performance of tcpreplay was conducted, a new version of the software has been released that enables more accurate packet scheduling. Therefore, in this subsection, the performance of tcpreplay version 2.3.3 – as studied in [128] – is compared to most recently released version 3.0 (beta 11). Throughout this subsection, the two versions are named tcpreplay v2 and tcpreplay v3, respectively. The major difference between the two – at least regarding their capabilities of accurately recreating packet flow traces – is the way the software behaves in between two successive packet transmissions. While tcpreplay v2 calculates the time to sleep until the next packet transmission, calls the nanosleep() function, and waits to be re-scheduled by the operating system, tcpreplay v3 utilizes a for loop and continuously calls gettimeofday() until the next transmission time is reached. While more CPU intensive, this is supposed to enable more accurate packet flow regeneration [140].

To investigate the difference in performance between tcpreplay v2 and v3, three separate packet flow regeneration experiments were performed for each of

the 5, 10, 15 and 20 Mb/s trace files. Each experiment lasted around 11 minutes. Table B.2 and B.3 show some relevant statistics of the performance of tcpreplay v2 and v3, respectively. From these tables, it can be observed that there are significant differences in performance between tcpreplay v2 and v3. The mean error in inter-packet spacing is generally lower for v3, as are the values for the 99^{th} percentile. For the 20 Mb/s trace regeneration experiments, 99 % of the inter-arrival times were within 0.04 ms of the intended spacing for tcpreplay v3, while the corresponding number for tcpreplay v2 is 1.16 ms. Thus, one may claim that tcpreplay v3 has better average performance than tcpreplay v2.

	Absolute difference in inter-arrival times		
Bit rate	Mean (ms)	99^{th} Percentile (ms)	Max (ms)
5 Mbps	0.139	1.058	5.62
10 Mbps	0.077	0.917	13.1
15 Mbps	0.073	1.061	12.9
20 Mbps	0.073	1.158	24.8

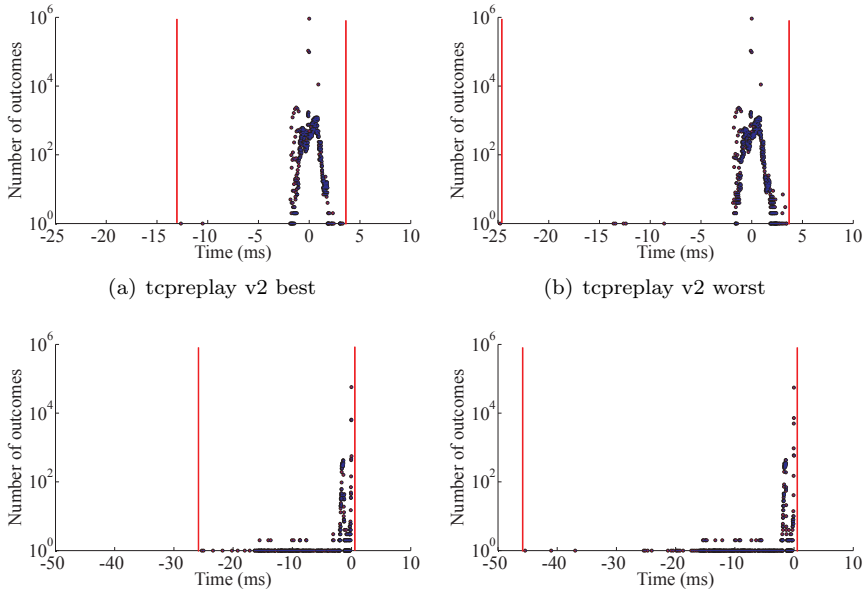
Table B.2: Statistics of the absolute difference in inter-arrival times between traces from 4-Sight and tcpreplay v2.

	Absolute difference in inter-arrival times		
Bit rate	Mean (ms)	99^{th} Percentile (ms)	Max (ms)
5 Mbps	0.007	0.013	33.4
10 Mbps	0.016	0.029	46.9
15 Mbps	0.021	0.032	34.2
20 Mbps	0.023	0.040	45.4

Table B.3: Statistics of the absolute difference in inter-arrival times between traces from 4-Sight and tcpreplay v3.

However, the worst-case performance, as indicated by the maximum difference in (absolute) inter-arrival time, is clearly worse for tcpreplay v3. Figure B.6 shows the best and worst results – among the three experiments – for tcpreplay v2 and v3 when regenerating a 5 Mb/s flow. While the re-scheduling mechanism employed in v2 tend to make the error in inter-packet spacing distributed around zero, the for-loop in v3 makes it is more likely that a packet is sent too late rather than too early. Hence, the distribution of differences in inter-arrival times between the tcpreplay v3 trace and the original 4-Sight trace exhibits a tail of negative values. Figure B.7 shows corresponding plots for the 20 M/s experiments.

Another way to illustrate the difference in worst-case performance, is to consider the number of packet inter-arrival times that are incorrect by more



(a) tcpreplay v2 best

(b) tcpreplay v2 worst

(c) tcpreplay v3 best

(d) tcpreplay v3 worst

Figure B.7: Distribution of difference in packet inter-arrival times comparing 20 Mb/s traces from tcpreplay v2 and v3 (Bin width equal to 0.01 ms)

than 2 ms. For the 20 Mb/s flow regeneration experiments, this number was 658 for tcpreplay v3, and 147 for tcpreplay v2. This translates to about 0.018 %, and 0.004 % of the total number of regenerated packets, respectively. Thus, one can argue that the greedy behavior of the packet scheduling algorithm in tcpreplay v3 results in a higher average, but a lower worst-case performance, than the re-scheduling approach used in tcpreplay v2. Lastly, note that these experiments were performed on a pc with a single-processor architecture. Running tcpreplay on a multi-processor machine is expected to provide a gain in performance, maybe especially for the CPU-intensive approach taken by tcpreplay v3.

Influence of clock drift

If the clocks on the packet regenerator and the packet capture device are not synchronized, then drift will occur when regenerating the packet flow. Figure B.8 illustrates this by showing the difference in arrival times between the original 4-Sight trace and the resulting trace regenerated by tcpreplay. Here, the data point (600,114) suggests that during a six-minute period, the clock in the packet

regenerator is drifting by around 114 ms relative to the GPS synchronized clock in the packet capturing device. This corresponds to roughly 16 seconds per day, which is considered to be common for desktop computers [149].

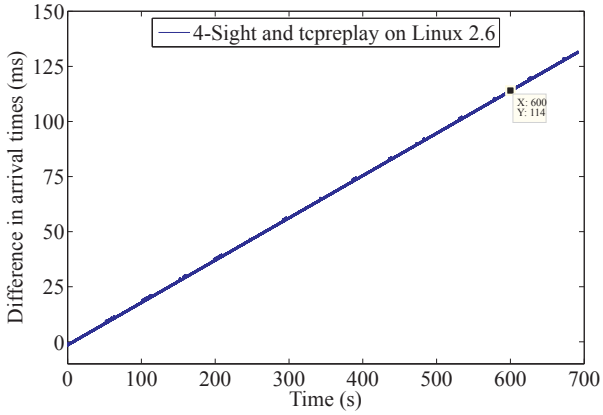


Figure B.8: Difference in arrival times between traces from 4-Sight and the tcpreplay packet regenerator

B.4 An Error Robustness Evaluation of H.264/AVC

This section considers using the proposed testbed to evaluate some error robustness aspects of the H.264/AVC video coding standard. The study and the results presented in the following were published in [130].

The section is organized as follows; Section B.4.1 gives a short overview of related work. Section B.4.2 describes the test setup and the different slice partitioning schemes to be compared. Finally, the resulting reconstructed video quality for different packet loss rates is discussed in section B.4.3

B.4.1 Related Work

As previously stated, the goal of any multimedia communication system should be to maximize the end-user's perceived quality of the delivered service. When evaluating the performance of such a service, either subjective testing, or some measure that is proven to show a high correlation to perceived quality, should be used to evaluate the performance of the service. Even though the peak signal-to-noise ratio (PSNR) has been shown not to correlate terribly well with perceived quality [94], it is still used in this part of the work. This is because of its simplicity and flexibility, but also due to the fact that still no widely recognized quality metrics exist, that incorporates the effect of packet loss and provides a higher correlation to perceived quality (see section A.1.1).

Most of the previously published work on H.264/AVC error robustness has been done using JVT's common test conditions for IP-based transmission [150], which includes a simple simulator discarding packets based on a loss pattern file. The loss patterns are obtained from Internet experiments, and with a few exceptions mainly consists of scattered packet loss [151]. In [20], Wenger gave a comprehensive overview of the error-resilience tools in the H.264/AVC standard, and presented results for six different encoder configurations comparing the use of intra macroblock updates, slice partitioning, interleaved FMO, dispersed FMO, and data partitioning for two CIF resolution sequences. FMO and data partitioning were concluded to be particularly valuable error resilience tools.

Halbach and Olsen [19] evaluated the use of motion-sensitive intra macroblock updates, i.e. areas with high motion are more likely to be intra-coded, as a means to stop error propagation. Stockhammer et al. [21] discussed the use of H.264/AVC in a wireless environment and presented an error robustness evaluation for a conversational service with and without feedback using the JVT common test conditions for RTP streaming over 3GPP [152]. While slice partitioning and rate-distortion optimized mode decision gave good results without the use of any feedback, excellent results were reported when multiple reference frames and feedback was used to effectively stop error propagation.

Calafate et. al. [153] studied the error resilience of H.264/AVC in an ad-hoc wireless network scenario.

While previous work largely has focused on low-resolution low-quality applications, this work studies the error robustness of a high-quality high-definition H.264/AVC broadcast service over an emulated IP network that introduces randomly distributed packet loss. We will not consider the delay and delay jitter introduced in real packet-switched networks, assuming that the playout buffer can be set large enough so that all delay jitter is absorbed. Although simple, random packet loss may be a valid model for low and transient congestion if queue management policies like Random Early Detection or Early Random Drop is employed in networks routers [154]. For typical Internet packet loss, more complex state models incorporating temporal dependencies of lost packets are often more appropriate [155].

One difference between this work and the above-mentioned contributions is that the proposed setup could easily be used to test and compare the error resilient encoding of real commercial H.264/AVC over RTP/UDP/IP broadcast solutions. Further, using the packet flow regenerator described in section B.2.4, the trace file from a given network emulation can be played back to a streaming media client for development, subjective testing, demonstration or other purposes. For instance, it could be used to recreate a specific lossy network condition for subjective testing of a streaming media session that has been impaired by packet loss.

B.4.2 Measurement Setup

Test Material

For our measurements we used an excerpt from the mini-movie "Standardized Evaluation Material" (StEM) under license from Digital Cinema Initiatives (DCI) [156]. More information on this clip is given in Annex I.1. The excerpt from StEM, in this section referred to as STEM-A, consists of 576 frames (from frame number 1716 up to and including frame number 2292) giving a playing time of 24 seconds. To create a 720p version of STEM-A, cropping was first employed to remove 768 pixels on both the left and right side of each frame, and also to remove 137 pixels from both the top and bottom of each frame. The resulting 2560 by 1440 pixel frames were then converted to the target 1280 by 720 resolution by bilinear downsampling.

Encoder Configuration

The STEM-A clip was encoded using the H.264/AVC reference software JM version 10.1 [145]. Table B.4 lists some important encoding parameters that were common to all configurations, unless otherwise stated, in Table B.5. IDR

(Instantaneous Decoding Refresh) pictures were used to prevent reconstruction errors propagating across GOP boundaries.

Parameter	*Value*
GOP structure	IPPP
Intra Period	24
IDR Intra Enable	On
Number of Reference Frames	5
Search Range	128
Symbol Mode	CAVLC
Weighted Prediction	Off
RD Optimization	On
MB Line Intra Update	Off
Random Intra MB Refresh	Off
Constrained Intra Pred	Off

Table B.4: Common encoding parameters for error robustness evaluation.

The eight different configurations we compared in this work are listed in Table B.5 together with the quantization parameter (QP) used in the encoding and the resulting average bitrate. "SL1" denotes the case when using no slice partitioning at all, that is, the entire primary coded picture is transported as a single NAL unit. Similarly, "SL5", "SL9" and "SL16" is encoded using 5, 9 and 16 slices per frame, respectively.

The JM reference software also supports creating slice partitions that are matched to the maximum transmission unit of the underlying network. All configurations having name starting with "SLMTU" are encoded in such a way that no NAL units are larger than 1450 bytes, which is the maximum transmission unit (MTU) payload on our Ethernet network, making sure that no NAL units are fragmented by the network abstraction layer. This feature, together with constrained intra prediction, which prevents intra-coded macroblocks from using inter-coded macroblock for prediction, is introduced in "SLMTU-CIP". Furthermore, "SLMTU-FMO-CIP" adds FMO using two slice groups per picture and the dispersed slice group map type, also known as the "checkerboard" FMO pattern [17].

While the above configurations have all been encoded using I and P frames only, and an intra period of 24 pictures, the next configuration instead uses groups of intra-coded macroblocks to refresh the prediction. In "SLMTU-FMO-MBLINE", one line of macroblocks, or rather group of macroblocks, are intra-coded in every picture. Considering that there are 45 lines of macroblocks

in a 720p frame, the number of regularly intra-coded macroblocks would correspond to an I frame about every 2 seconds for our test material. The last configuration, "SLMTU-FMO-PYR", was encoded using a four level hierarchical coding structure as discussed in section 2.1.1.

Name	*QP*	*Bitrate*
SL1	26	2950 kbps
SL5	26	3019 kbps
SL9	26	3078 kbps
SL16	26	3178 kbps
SLMTU-CIP	27	2910 kbps
SLMTU-FMO-CIP	28	3016 kbps
SLMTU-FMO-MBLINE	28	3034 kbps
SLMTU-FMO-PYR	27	2990 kbps

Table B.5: The eight configurations used in H.264/AVC error robustness testing.

It is well known that extensive slice partitioning decreases compression efficiency, since prediction is constrained within one slice of a picture and also because of the increased overhead associated with slice and protocol headers [19]. Figure B.9 shows the rate-distortion performance of the different configurations (without considering protocol overhead). The quantization parameter was fixed for all pictures in the sequences and was chosen so that the average bitrate of all configurations was as close to a target rate of 3 Mb/s as possible. We observe that for our test material, the dispersed FMO scheme decreases the compression efficiency by roughly 1 dB for the target rate. This is due to the restriction FMO places on the use of intra-picture prediction techniques [20].

Transmission and Network Emulation

The eight H.264/AVC video files output from the JM reference encoder were wrapped in MP4 files and hinted with an MTU of 1450 bytes using tools from MPEG4IP [147]. The media was then broadcasted over the test network, through an Empirix PacketSphere IP network emulator [136], and captured by a high-performance DAG network interface monitoring card [138]. The network emulator was configured to introduce uniformly distributed packet loss at different rates, namely 0.01%, 0.1%, 0.25%, 0.5%, 1%, 1.5%, 2%, 3% and 5%. Each media file was transmitted five times over the test network at each configured packet loss rate, giving a total of 45 measurements for each of the eight configurations.

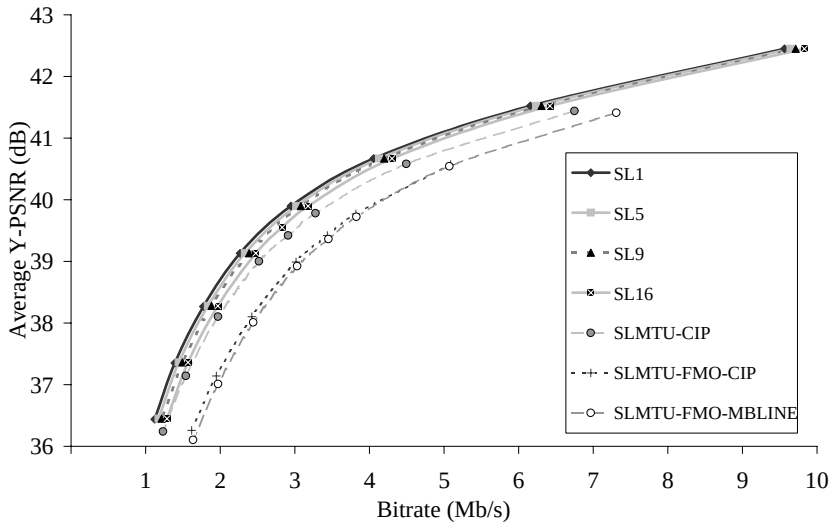


Figure B.9: Rate-distortion performance of the different configurations used in error robustness testing

Stream Assembly and Decoding

Figure B.10 summarizes the test setup. Due to the high complexity involved in real-time decoding of High-Definition H.264/AVC, and because of the current limitation in available real-time decoding software supporting the advanced error resilience tools available in H.264/AVC Extended Profile, packet reception and decoding was done offline. A purpose-built application, *pcap2avc*, was developed to perform the packet reception and stream assembly, as described in Section B.2.6. Finally, decoding of the received bitstream was done using the H.264/AVC reference software [145].

JM version 10.1 supports the ability to conceal entire frame losses, which is particularly helpful for configuration "SL1", where the loss of a single FU-A NAL Unit means that the entire frame is lost. The current limitation of the picture error concealment algorithm in JM 10.1 is that it can only conceal I and P pictures, and that it is not able to conceal the last P picture of a GOP if this is lost. This has to be taken into account when calculating the PSNR to prevent misalignment between frames in the original and decoded sequences. As long as at least one slice is correctly received for a picture, the JM reference software is capable of concealing lost slices using temporal and spatial error concealment algorithms [157].

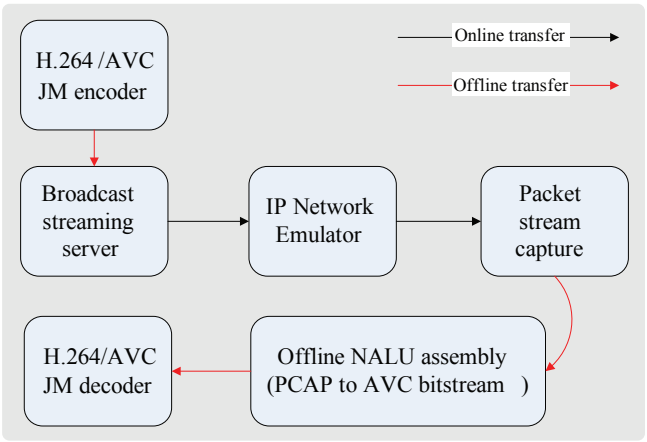


Figure B.10: Measurement setup for error robustness evaluation of H.264/AVC

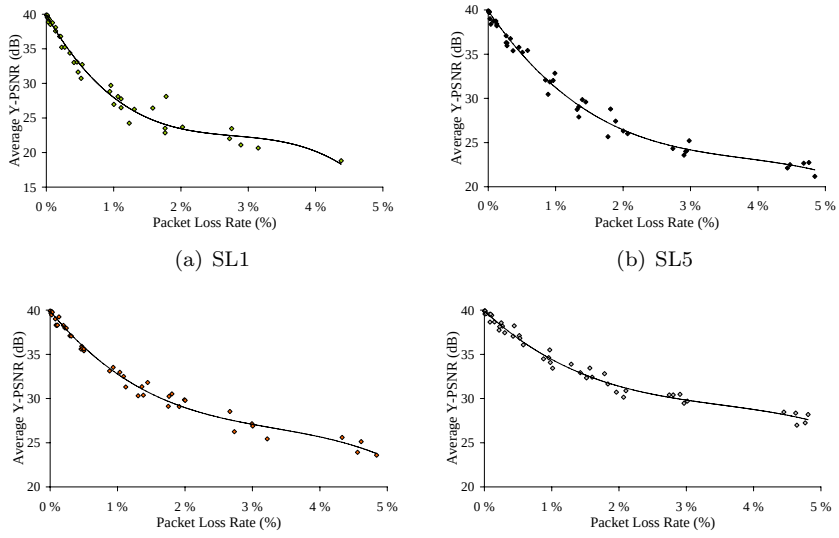
B.4.3 Results

In this section we present results of the reconstructed video quality for the eight different configurations described in section B.4.2. The peak signal-to-noise ratio (PSNR) between the uncompressed original and the decoded video is calculated based on the luminance video component only, denoted Y-PSNR. When calculating the Y-PSNR, no penalty was introduced for missing, non-concealed frames.

Figures B.11 and B.12 show reconstructed Y-PSNR at packet loss rates (PLR) from 0.01% to 5%. One dot in each of the scatter plots corresponds to the average Y-PSNR across all the frames of the reconstructed video sequence for a single run through the testbed. Curves estimated using second or third order polynomial regression (depending on which order made the best fit) are also shown in the graphs for purpose of illustration.

From the plots we can clearly see that the error resilience with respect to reconstructed Y-PSNR improves significantly when more advanced slice partitioning schemes are being employed. For instance, if we compare at which packet loss rate the configurations in Figure B.11 seem to have quality reduction to 35 dB Y-PSNR on average, we see that "SL1" reaches this point at a loss rate of approximately 0.3%. The corresponding figures for "SL5", "SL9", and "SL16" are 0.5%, 0.6%, and 0.9%, respectively.

When the slice partitioning is matched to the MTU of the underlying network, and constrained intra prediction is employed, we see from Figure B.12(a) that the



(a) SL1 (b) SL5

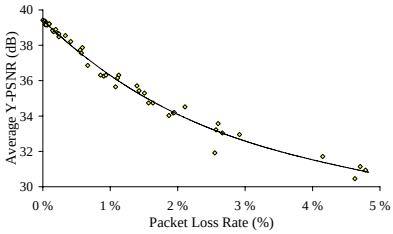
(c) SL9 (d) SL16

Figure B.11: Average reconstructed Y-PSNR as a function of packet loss rate for different encoder configurations.

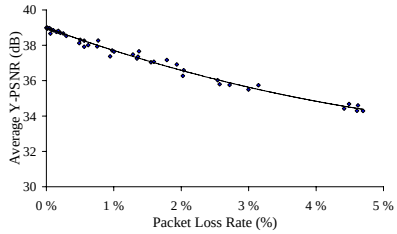
performance is further improved. For "SLMTU-CIP", the reconstructed Y-PSNR stays above 35 dB for packet loss rates up to 1.5%.

Figure B.13 shows how the deficiency in proper slice partitioning affects the video delivery when packet loss occurs. The effective packet loss rate, here equal to the total number of transmitted packets divided by the sum of lost and discarded packets, is plotted as a function of packet loss for five sets of measurements. For "SL1", where no slice partitioning is employed, nearly 20% of the packets have to be discarded when 3% of the packets are lost in the network. The situation improves with increasing slice partitioning, and for "SLMTU-CIP", no packets have to be discarded, meaning that the effective PLR is equal to the PLR inflicted by the network.

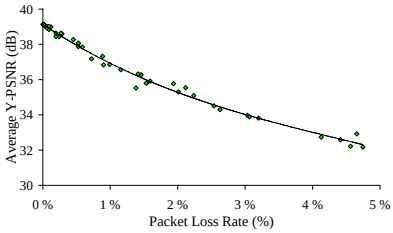
From Figure B.12(b) we can clearly see what can be gained by using FMO for moderate and higher loss rates. While the reconstructed Y-PSNR for "SLMTU-CIP" decreases below 35 dB at around 1% packet loss, "SLMTU-FMO-CIP" can maintain an average Y-PSNR above 35 dB for up to 3.5-4% of random packet loss for our test material. Also note from Figures B.12(a) and B.12(b) that, while decreasing the compression efficiency by 0.5 dB in the error free case, the FMO-based scheme seems to outperform the pure slice partitioning based scheme at packet loss rates higher than $0,25\%$.



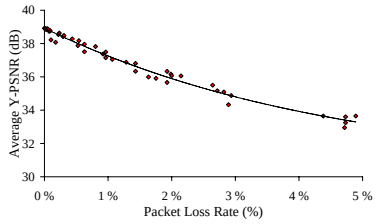
(a) SLMTU-CIP



(b) SLMTU-FMO-CIP



(c) SLMTU-FMO-CIP-PYR



(d) SLMTU-FMO-CIP-MBLINE

Figure B.12: Average reconstructed Y-PSNR as a function of packet loss rate for different encoder configurations.

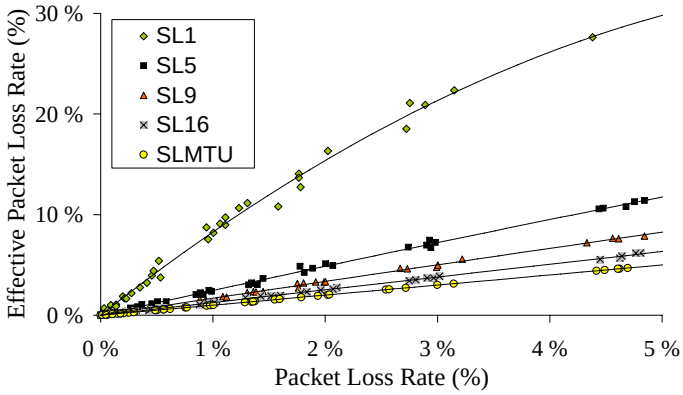


Figure B.13: Effective loss (lost and discarded packets) as a function of packet loss rate for different slice partitioning configurations

The measurements for "SLMTU-FMO-CIP-MBLINE" reveal that intra coding a single group of macroblocks in every picture gives an error robustness performance comparable to that of using I pictures for random packet loss. In Figure B.12(d) we see that the reconstructed Y-PSNR is decreasing just slightly faster than for "SLMTU-FMO-CIP" in Figure B.12(b), and that it stays above the 35 dB line for packet loss rates up to 2.8%.

Interestingly, we see from Figure B.12(c) that the hierarchical coding scheme using B pictures in "SLMTU-FMO-CIP-PYR" actually has reduced error robustness compared to "SLMTU-FMO-CIP", and reconstructed Y-PSNR falls below 35 dB for PLR higher than 2.5%. This indicates that, unless some prioritization mechanism is in place to protect the all-important base layer (consisting of the I and P pictures in the pyramid), then the $IPPPP$ coding structure of "SLMTU-FMO-CIP" is superior to the pyramid coding in "SLMTU-FMO-CIP-PYR" in terms of error robustness. In the error free case however, pyramid coding improves the compression efficiency by 0.2 dB for our test material when FMO is used (see Figure B.9).

B.5 Summary and Discussion

We have presented a streaming media testbed for IP networks. Besides a streaming server and a streaming media client, it consists of an IP network emulator, a high-performance packet capture device and a packet flow regenerator enabling repeatable performance measurements of streaming media applications. Analyzing the delay introduced by the packet regenerator in our measurements, it was observed that this could be limited to around ± 2.0 ms for a 5 Mb/s flow when tcpreplay was running on a Linux 2.6 kernel. Depending on the application at hand, this uncertainty may or may not be significant. For instance, interactive conversational applications require small playout buffers (in the millisecond region), and thus this uncertainty introduced by the packet regenerator is more significant than if the playout buffer is large (in the region of several seconds), like in video-on-demand or multicast streaming scenarios.

Further, using the proposed testbed, an evaluation of the performance and error robustness of an RTP-based high-quality H.264/AVC video broadcast streaming service was presented. IP network emulation was used to introduce a varying amount of randomly distributed packet loss. A high-performance network interface monitoring card was used to capture the video packets, and the RTP packets in the resulting trace file were assembled to an H.264/AVC Annex B byte stream file, which was then decoded by the JVT JM 10.1 reference software. The proposed measurement setup represents a practical and intuitive approach, and can easily be used to perform error resilience testing of real-life H.264/AVC broadcast applications.

Through a series of experiments, some of the error resilience features of the H.264/AVC standard was evaluated for packet loss rates between $0,01\%$ to 5%. The results confirmed that an appropriate slice partitioning scheme is essential to have graceful degradation of application behavior in the case of packet loss. While FMO reduces the compression efficiency about 1 dB for the test material used, reconstructed video quality is improved for loss rates of 0.25% and higher. At loss rates up to 3-4% for our test material, flexible macroblock ordering together with slice partitioning matched to the MTU of the underlying network can give remarkable both objective and subjective visual quality.

In the error robustness study described above, the only objective metric used for assessing the impact of packet loss was the average sequence PSNR. This metric does not necessarily reflect end-user perceived quality very well, and therefore, further validation using state-of-the video video quality metrics and formal subjective testing should be carried out.

Further work could also include extending the measurement framework so that both packet capture and NAL unit assembly could be done in real-time using MAPI [146].

Part C

Adaptive Video Streaming

Part C

Adaptive Video Streaming

This part presents a network-adaptive video streaming system that employs scalable video coding techniques and a media-friendly congestion control scheme to ensure that end-users are provided the best possible experience when using a video-on-demand service. Standard solutions for media encoding (H.264/MPEG-4 SVC) and delivery (RTP) are used to build a prototype system. A specific application scenario is simulated and evaluated, namely video distribution over IEEE 802.16 broadband wireless networks.

C.1 Introduction

It is anticipated that many future multimedia systems and applications will be deployed in a ubiquitous manner; end-users may have terminals with varying capabilities, and either wired or wireless network connectivity. The different usage scenarios present different challenges with respect to the delivery of multimedia content. For instance, bandwidth is a much scarcer resource in wireless networks than in their wired counterparts. Due to this reason, the requirement that multimedia applications should be able to adapt to varying network conditions is most challenging in a wireless context.

IEEE 802.16-based technologies are currently the subject of much interest within the community: Operators are rolling out systems compliant with the current versions of the standards and starting to offer services, while the research community is focused on developing new capabilities for such systems. Much of the current research focus is on the development of mobile and mesh variants of the technology. However, there is still work to be done on understanding how the more established fixed broadband wireless access (FBWA) variant performs for particular applications.

Many are of the opinion that wireless technologies in general and 802.16-based technologies in particular are a natural solution for connectivity within rural areas as they scale well and can deliver high data rate wireless connectivity at reasonable cost. Further, there are typically less radio propagation problems in rural settings than there are in built-up urban or suburban environments. For these reasons, it is interesting to focus on the capabilities of 802.16 technology in the context of last-mile rural broadband access [158–162].

Those experimenting with 802.16 technologies typically have a number of issues they wish to investigate: some of these issues relate to the radio propagation aspects of the technology, some relate to the throughput that the system can deliver in different configurations and some relate to the quality that can be offered to different applications. Regarding this latter activity, much of the focus has been on the capability of 802.16 systems to support VoIP – little to date has been reported on how 802.16 systems can be used to support video services.

Video distribution over IP networks is also the subject of much interest at present, in particular in the IPTV arena. IPTV includes the transmission of both live television programmes and events, and Video on Demand (VoD) services over broadband access networks. Increased penetration of higher capacity broadband access and more efficient video compression schemes are making distribution of high quality video content over an IP-based infrastructure a realistic option. One particularly interesting technology that is being developed within this context is the scalable extension to H.264/AVC, so-called Scalable Video Coding (SVC) [25]. SVC can work particularly well with the adaptive mechanisms that are necessary for distributing video over resource-constrained networks.

C.1.1 Related Work

An interesting contribution which focused on the general challenges associated with adaptive video streaming in a wireless context was made by Wu et al. [79]. There, they identified three components essential to delivering acceptable video in such scenarios: scalable video representations, end-systems capable of performing network-aware adaptation, and adaptive QoS support from the network. The solution proposed here has support for these three components, although the QoS support from the network is not studied. While Wu et al. identify a high level framework for streaming video over wireless networks, there are many details which are unspecified in their work and hence more work is needed to address this.

The related contributions can be categorised into those focused on performance of 802.16 systems, those considering issues with adaptive streaming of scalable video, or those addressing congestion control issues for video traffic. Each of these is dealt with in the following subsections.

Performance of IEEE 802.16 Wireless Broadband Networks

A number of contributions have been made which focus on different aspects of the performance of 802.16 systems. Ghosh et al. [160] gave one of the first reports on the performance of 802.16a. They identified limitations of the technology and pointed out that the performance that can be anticipated by the early realizations will be significantly lower than many had thought. However, they did acknowledge that there are well-known techniques, which can be employed to significantly improve the system performance, such as MIMO and adaptive subcarrier loading.

A very important contribution to this area was made by Hoymann in [161], in which he provides some insight regarding the data rates that can be achieved for 802.16d/802.16-2004 systems operating under different assumptions. His results indicate that cell throughput on the order of 10-20 Mb/s is attainable for realistic spatial distributions of users. Further, he highlights the inefficiencies arising from padding of OFDM symbols and he also concludes that the overhead introduced by the MAC headers is about 10 %.

Cicconetti et al. [158] examine the performance of the 802.16 MAC for two scenarios — one based on residential users and one based on small and medium-sized enterprise (SME) users. They focus on how the system performs for delay sensitive applications and report useful results which indicate the capacity of the system for various load configurations. More specifically, they envisage providing services to something on the order of a few 10s of users simultaneously using a channel configured according to one of the recommended WiMAX profiles [163]. Further, they point out the increasing overheads associated with increasing the numbers of subscribers. Lastly, in [164] the authors consider the throughput of 802.16e systems. While their work is in a mobility context, they consider slow moving users and efficient modulation and coding schemes. Hence, their work can be used as an indicator of the types of data rates possible in 802.16 systems. More specifically, the results indicate that a single cell can deliver a little over 11Mb/s aggregate data rate under some quite realistic assumptions.

Streaming of H.264/MPEG-4 SVC

The forthcoming SVC standard will specify how to represent video streams with spatial, temporal and quality scalability; indeed it is possible to generate SVC enhancement layers which can augment an H.264/AVC compatible base layer [25]. In the SNR quality dimension, both layered coarse-granular scalability (CGS), and fine-granular scalability (FGS) is supported. The focus here is on the FGS capabilities of the standard[1].

[1] Please consider comments made in section 2.1.2

There is much published work on streaming of H.264/AVC video, and the performance of the wide range of error resilience tools available in the standard, for both wired, wireless, and mobile scenarios, see e.g. [165], [21], and [20]. Schierl et al. [166] describes using SVC in combination with an unequal error protection based scheme for wireless broadcasting, while Nguyen and Ostermann [167] presented the first SVC-based adaptive solution for Internet streaming, employing packet-train techniques to estimate available bandwidth. Recently, Wien et al. presented a real-time adaptive streaming video system based on MPEG-21 and SVC in [168]. However, to date, little has been published on the use of H.264/AVC or SVC in an 802.16 context.

Congestion Control for Scalable Video Streams

As mentioned in Section 2.2.3, congestion control is an integral part of any adaptive media streaming solution. While congestion control for video streaming applications is quite well-studied, there are still some open issues. In [42], Vieron and Guillemot address the rate smoothness and real-time requirements for video streaming, in the case of the standard TFRC protocol [52]. There are some issues with this approach: the sending rate still fluctuates substantially which is undesirable and the scheme relies on packet loss to perform efficient congestion control. While video applications can be designed to have resilience against lost packets, it is advantageous to minimize the error rate when the system is operating in steady state.

Given sufficient buffer space in network routers, accumulation-based protocols – as discussed in Section 2.2.3 – can operate without packet loss in the steady state. Further, it is well-known that the performance of protocols that rely on packet loss to perform congestion control – such as TCP and TFRC – perform poorly in wireless radio networks where loss also occurs due to errors on the wireless channel [50, 169]. Since accumulation-based protocols do not utilize packet loss for detecting congestion, they are a favorable candidate for transport in wireless networks. Hence, an accumulation-based control was chosen in this work. More details on the specifics of the scheme will be presented in section C.2.3.

It is worth noting that there are some issues with such schemes, specifically related to reverse-path congestion. Xia et al. have shown that these can be solved by using a high-priority channel for special control packets and appropriate receiver side behavior. However, in the work described here, this issue does not arise as there are sufficient resources allocated to the reverse path. Consequently, network support for QoS is not necessary, and a pure end-to-end approach is taken.

C.1.2 Outline and Credit

The focus of this work, is on studying issues associated with video delivery over 802.16 FBWA networks. More specifically, the focus is on determining how well SVC - in conjunction with specific congestion control algorithms - works for distributing video to Subscriber Stations (SSs) of an 802.16 system. Questions to be answered also include how the adaptive video mechanisms impact radio resource utilization and how many users can be supported for different system configurations. The setting is assumed to be rural, as this is a natural use-case for 802.16 systems: clearly, this has implications for the spatial distribution of users in the system.

The structure of Part C is as follows. An overview of the system under study is presented next in section C.2. In section C.3, the particular scenarios simulated are described, and the results obtained are presented and discussed. The work is concluded in section C.5.

The work presented in this part is based on a close collaboration with the Performance Engineering Lab at University College Dublin, Ireland. More specifically, the 802.16 network simulation model was developed by Vasken Genc, while the rate-adaptive video server and client applications were developed by the author. Further, the project was performed in close consultation with Dr. Seán Murphy, who also contributed invaluably in writing the corresponding papers [170] [171].

C.2 System Overview

This section will present an overview of the video distribution system, a model of the 802.16 wireless network that is used for delivery, and the proposed adaptive video streaming solution.

C.2.1 Video Distribution Architecture

The system architecture assumed, as depicted in figure C.1, is one in which users access video content from a video server located within, or close to the service provider's core network. While a single server, in general, will not have access to all video content users might request – this could be for copyright or storage reasons, for example – it may be reasonable to envisage a server in the distribution network which stores popular content. This could be part of a caching mechanism in a content delivery network, or it may occur due to some agreement between the access provider and the content provider. Also, the architecture assumed is a realistic starting point for simulation studies as most of the congestion and delays occur on the wireless access link.

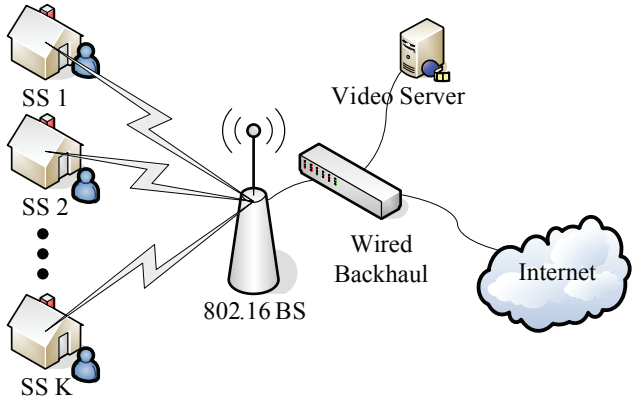


Figure C.1: System architecture

In the envisaged architecture, the video applications – i.e. the streaming server, and the streaming clients on subscriber stations SS 1 to SS K in figure C.1 – are assumed to be adaptive so as to enable them to make best use of the available resources in the 802.16 system. More specifically, the adaptive mechanisms enable each flow to increase its rate if the radio access system is under-utilized or, conversely, each flow reduces its rate if the system enters congestion.

It is important for the perceived quality of video-on-demand services that no users experience interruptions in continuous media playback. These interruptions occur when the streaming client's playout buffer drains completely and happens because of severe and persistent congestion in the network. Therefore, limiting the amount of data that is in the network, as accumulation-based congestion control mechanisms do, can, in effect, also help prevent the client's playout buffer from draining during a streaming session.

To facilitate this adaptation, the video content is encoded in such a way as to provide spatial, temporal or quality scalability. How to decide on the optimal adaptation procedure, with respect to which combination of scalability dimensions to choose at any particular time, is in itself an important and difficult research problem. Here, we take the view that scaling the video in the quality/SNR dimension when network conditions deteriorate should be taken as the first course of action. Since we use the fine-granular scalability (FGS) configuration of SVC, the packets of the FGS quality layer can be truncated at an arbitrary point to perform the bit rate adaptation. Later, if congestion persists, temporal layers should be skipped to further reduce the transmission bit rate, in effect reducing the frame rate with a certain factor for each temporal layer being removed. However, this mechanism should only be employed rarely

as users typically may find it perceptually unacceptable.

It should be noted that the use of scalable coding, and perhaps particularly fine-granular scalable coding, comes at the expense of lower compression efficiency as compared to single-layer coding. However, one of the objectives of this work is to show the advantages of such schemes, despite the added redundancy. Two obvious strengths that are important for delivery in wireless scenarios are 1) the ability to adjust the transmission rate when congestion occurs without relying on transcoding, and 2) the error resilience aspects of combining the important base layer with some unequal error protection scheme.

In this study, all the traffic going through the 802.16 base station originates from the video applications. The impact of TCP traffic was not considered here. This is because previous studies of TCP Vegas – which can be considered to be an accumulation-based congestion control protocol [58] – have shown that Vegas does not receive a fair share of the bandwidth when competing with variants of TCP that infer congestion solely through loss, such as TCP Reno [172]. Therefore, one may argue that a real-life system employing the proposed adaptive video solution would have to separate video traffic from other types of traffic, using e.g. differentiated services [76].

C.2.2 Network Simulation Model

The simulation was performed using the NCTUns simulator [173]. This simulator was chosen as it enables real video streaming applications to interact easily with the simulation tool. More specifically, the video application traffic generators were actual streaming clients and servers – rather than simulation models – that could input traffic into the simulator. The simulator was modified so as to provide support for IEEE 802.16 [170].

An overview of the NCTUns simulator is given next. This is followed by a description of the 802.16 simulation model that was added to the NCTUns simulator.

The NCTUns Network Simulator

One of the key design issues with NCTUns [173] was that it was intended to provide support for interaction between real applications and a network simulator. To this end, the simulator comprises of functionality within the Operating System (OS), which can capture packets generated by an application and route them to a simulation entity. More specifically, the applications use the UNIX TCP/IP protocol stack on the machine where the simulator is installed, and transmit packets to a virtual network interface called a tunnel. The NCTUns simulation kernel captures the packet from the source virtual interface, processes it, and transmits the packet to the destination virtual interface. The application on the

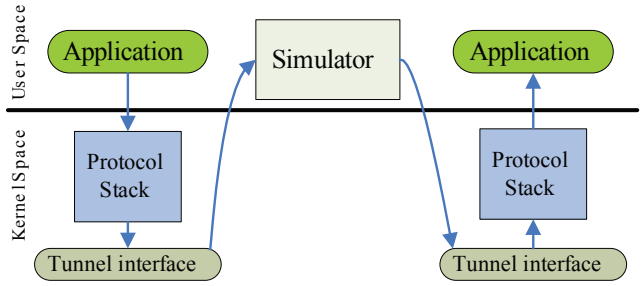


Figure C.2: Architecture of the NCTUns simulator

destination host then receives the packet through the TCP/IP protocol stack just as if it were communicating with the source application over any real network. A simple view of the operation of the simulator is given in Figure C.2.

IEEE 802.16 Network Model

The 802.16-2004 standard comprises of a definition of multiple PHY layers and a single MAC layer; aspects of both the PHY layer and MAC layer were modeled in the simulator. However, the PHY and MAC definitions are quite closely coupled in 802.16 and consequently, it is difficult to make a clear separation between them in a simulation model. The simulation model that was developed incorporates the following aspects of the 802.16-2004 standard:

- The OFDM PHY (Orthogonal Frequency-Division Multiplexing) was used, as most shipping 802.16 systems use this PHY;
- half-duplex TDD (Time Division Duplex) was assumed as this results in lower cost equipment;
- modeling of Downlink (DL) and Uplink (UL) subframes and control over how much resources are allocated in each direction;
- modeling transmission of Frame Control Header (FCH);
- modeling a subset of the 802.16 burst profiles.

Each frame is transmitted as follows. Firstly, the FCH is transmitted. This is followed by the transmission of the DL subframe, which is divided into transmissions for each of the different so-called burst profiles. A burst profile defines the modulation and coding schemes used in the transmission. For each

profile, a queue is checked to see if any packets need to be transmitted: if some packets need to be transmitted, they are transmitted using FCFS (First Come First Served) until either the buffer empties or there is no capacity remaining for this burst profile in this frame. Once the DL subframe has been transmitted, the UL subframe starts and subscriber stations are provided access to the medium. [2] If a packet does not fit the remaining OFDM symbols in a frame, it is fragmented. The implemented mechanism only fragments the last PDU (Packet Data Unit), since it is shown in [161] that fragmentation of all PDUs does not provide any gain in terms of MAC frame utilization due to the extra overhead.

In this work, the resources are allocated such that each subscriber station can receive approximately the same bitrate. This is done by allocating resources to each burst profile relating to the number of subscriber stations using that burst profile and the transmission efficiency of that burst profile (in terms of bits/symbol). Thus, the number of physical slots allocated to each burst profile on the downlink can be written as

$$PS_k = PS_{tot} \cdot \frac{SS_k \cdot W_k}{\sum_{k=1}^{k} SS_k \cdot W_k}$$

where, PS_{tot} denotes the total number of physical slots allocated to a frame, SS_k is the number of subscriber stations using burst profile k, and W_k is primary weight in Table C.1.

Table C.1: Resource allocation between burst profiles

Burst Profile k $K = 7$	Bit/symbol	Weight W_k	Example: $PS_{tot} = 100$	
			SS_k	PS_k
BPSK 1/2	1/2	9	1	38
QPSK 1/2	1	9/2	1	19
QPSK 3/4	3/2	3	1	13
16QAM 1/2	2	9/4	1	9
16QAM 3/4	3	3/2	1	7
64QAM 2/3	4	9/8	1	6
64QAM 3/4	9/2	1	2	8

Note that the resources allocated to each burst profile are shared. More specifically, each burst profile had a single dedicated buffer in the BS which could be used by all nodes using that profile: any packet destined to one of these nodes

[2] Note that only one subscriber station is transmitting at a time, and that their individual transmission schedule is decided by the BS.

would be inserted into the appropriate buffer. In this way, the resources available for a single burst profile were shared. This approach was used rather than a more sophisticated scheme with separate per-flow queuing as it is considerably simpler and could result in some cost savings.

The modulation scheme used between the BS and each SS is determined by estimating the signal to noise ratio (SNR) at the receiver and mapping this to one of the thresholds specified in 802.16-2004 standard [174]. More specifically, the receiver SNR is first computed using the approach of Hoymann [161]. This is then mapped to an appropriate modulation and coding scheme using the indicative values relating received SNR to appropriate modulation and coding scheme provided in the standard.

C.2.3 Adaptive H.264/SVC Streaming System

This section will present the adaptive video distribution system. In fact, two different rate adaptation schemes have been considered in this project. In the first scheme that was developed, the amount of enhancement-layer data to transmit is adapted according to the occupancy of the receiver playout buffer, and was described in [170]. This leads to a system in which end-users experienced a similar risk of buffer underflow and interruption in continuous playback. However, the adaptation scheme is somewhat agnostic with respect to transmission rate, and seemed too slow in reacting to congestion, so an alternative scheme was investigated.

The second scheme uses ideas from the concept of accumulation-based congestion control to decide on the proper transmission rate, and is documented in [171], and also described later in section C.2.3. In order to make the presentation in this part more coherent, here the focus is on the latter adaptation scheme. The reader is referred to [170] for a detailed description of the former approach, and corresponding simulation results and discussion.

Operation of the Streaming Media Server

The Joint Video Team (JVT) working group of ISO/IEC and ITU implements and maintains reference software for SVC called JSVM (Joint Scalable Video Model) [175]. The JSVM includes a tool which extracts specific scalable layers from an encoded SVC bitstream, and was thus taken as starting point for the streaming server in this project.

RTP/UDP [31] was used for transporting the H.264/AVC and SVC Network Abstraction Layer (NAL) units. Fragmentation of NAL units larger than a specified Maximum Transmission Unit (MTU) was performed according to the draft specification of the RTP payload format for SVC [38]. The different fragments were constructed to have equal size, putting any remaining bytes into

the first fragment. No aggregation of NAL units was performed, and the NAL units were sent in decoding order assigning increasing timestamp values to the access units as they were sent. Here, an access unit is the set of NAL units and possibly Supplemental Enhancement Information (SEI) NAL units belonging to a particular temporal level in the scalable bitstream.

Operation of the Streaming Media Client

At the receiving side, the client parsed the RTP packets, calculated the packet decoding deadline, updated reception statistics, and inserted the packets into a playout buffer. There was an initial pre-buffering period, after which, all packets belonging to an access unit were extracted from the playout buffer when that access unit's deadline was reached. The corresponding NAL units were then assembled, and the correctly received NAL units were written to an Annex B byte stream file [14]. Decoding was later performed offline using the JSVM reference software.

Proposed Congestion Control and Rate Adaptation

As mentioned in Section 2.2.3, when the estimated accumulation $a_i(t)$ of a flow i is low, the sending rate is increased; conversely, if $a_i(t)$ exceeds some target accumulation a^*, then the sending rate is reduced to drain the excess accumulation.

Using the same notation as in [58], the congestion control mechanism can be summarized in the time domain by the following equations:

$$\dot{w}(t) = -g(t) \cdot (a_i(t) - a^*) \tag{C.1}$$

where

$$g(t) = \begin{cases} \frac{\kappa \cdot rtt_{ip}}{rtt_i} & \text{if } a_i(t) \leq a^* \\ \frac{\kappa}{rtt_i} & \text{if } a_i(t) > a^* \end{cases} \tag{C.2}$$

In Equation C.1, $\dot{w}(t)$ denotes the change in congestion control window, κ is a positive congestion control constant, while rtt_i and rtt_{ip} denote the round-trip time and the round-trip propagation delay, respectively. Including the fraction rtt_{ip}/rtt_i will limit the overshoot when increasing the rate, while having a proportional decrease in equation C.1 will ensure a faster reaction time. Clearly, a^* determines how much data can be in the network at any time. κ determines how reactive the system is.

Due to the nature of scalable encoded video, adaptation of the transmission rate is best performed at the start of a GOP, i.e. immediately before transmitting

a I/P picture of the next sub-stream. Therefore, the congestion window value is adjusted at these key instants based on the most recent available feedback.

The NAL units of an H.264/SVC bitstream do in general have widely different size. Since a small packet would cause less congestion on the wireless access link than a large packet, the congestion control mechanism should operate on the number of bytes instead of the number of packets.

The congestion window is adjusted once every GOP, resulting in the target number of bytes for the following GOP, $W(t)$. However, a pre-encoded scalable video stream places some restrictions on the value and range of $W(t)$. Firstly, there is an minimum amount of data that must be sent, $W_{\min}(t)$, corresponding to the base layer I/P picture. Secondly, there is a maximum amount of data, $W_{\max}(t)$, which corresponds to transmission of all scalable layers. It is possible that $W(t)$ as calculated using equation (C.1) falls outside this range. As it is not possible to transmit less data than $W_{\min}(t)$ for the GOP without having to skip transmission for the entire GOP (with negative implications for the decoder until the next I picture), the congestion window is not permitted to fall below $W_{\min}(t)$. Hence, if the calculation of $W(t)$ results in $W(t) < W_{\min}(t)$, then $W_{\min}(t) \rightarrow W(t)$. Alternatively, if there is insufficient data to fill the congestion window and $W(t) > W_{\max}(t)$ then $W(t)$ remains unchanged.

The actual video rate adaptation is performed in the following way; first, considering only a single spatial layer, the appropriate number of temporal levels for the quality base layer is found, for the given value of $W(t)$. If the combined size of all temporal levels is still lower than $W(t)$, the remaining part of the transmission budget is filled with FGS quality enhancement data. This is done so as to ensure that an equal fraction of FGS data is transmitted for each picture in the GOP. If only a subset of the temporal levels can be sent, the FGS layer packets corresponding to these levels are cropped to meet the target rate. In this way, the algorithm is able to meet the target rate perfectly.

This policy of giving preference to frame rate over SNR quality was chosen because, especially for video sequences with moderate and high motion, people tend to favour keeping a high frame rate [85]. For a discussion on the relationship between video acceptability and frame rate, see [176].

The next challenge is how to estimate accumulation – the amount of flow data that is in transit in the network. This is done at the server side using standard RTP protocol feedback mechanisms [31]. In all RTCP receiver reports, clients inform the sender of the highest RTP sequence number they have received (HSNR). At the sender side, the server keeps track of the highest sequence number sent (HSNS), together with S_n, the size of packet n. Accumulation, in bytes, is

then estimated according to the following formula:

$$a_i = \sum_{n=HSNR+1}^{HSNS} S_n$$

Accumulation-based controls are developed from analytical models that assume fluid properties of the flows in the network, i.e. packets are infinitely small and divisible. When employing such a scheme for a highly bursty video application with variable packet sizes, care has to be taken when deciding on the proper packet scheduling. To this end, a simple approach was used in which the transmission rate in a GOP is made as smooth as possible. This is done by introducing inter-packet spacing which was proportional to packet size, similar to that described in [52].

For every GOP having a target rate of R_{GOP}, the transmission interval t_{int} between packet n and $n+1$ in that GOP is calculated as:

$$t_{int} = S_n / R_{GOP}$$

Due to limitations in timing granularity and scheduling in common operating systems, a slightly modified approach should be used in a real-world implementation. If the calculated time until the next packet transmission is very small, e.g. half of the timer granularity, the packet is sent immediately. Chapter 4.6 of the TFRC specification [52] provides a good discussion on this topic. This modified scheduling behaviour was included to better reflect how a real-world system could operate.

It is worth noting that round-trip-time was estimated according to [31]. Thus, the value of rtt_i was an exponentially weighted moving average (EWMA) of the RTT samples, using a smoothing factor of 15/16. Finally, the minimum observed RTT sample value was used as an estimate for the round-trip propagation delay rtt_{ip}.

C.3 Simulation Results

A number of simulations were performed to study different aspects of the behaviour of the system. The system capacity was investigated, as was the performance of the adaptive video streaming algorithm and the resulting video quality. Each of these are described in the subsections below. These are preceded by a more detailed description of the simulation scenarios.

C.3.1 Simulation scenario

There are two distinct aspects to the simulation scenarios: issues pertaining to the configuration of the wireless access network and those relating to streaming of video. The network configuration is discussed first, followed by a discussion of the video issues.

In the scenarios studied, the IEEE 802.16 system was configured to operate in Point to Multipoint (PMP) mode. The OFDM PHY was assumed, with 200 subcarriers used for data. The channel bandwidth was 20 MHz, and a cyclic prefix value of 1/4 was chosen as this results in the most robust system. Finally, the system was configured such that 80 % of the capacity was for use in the downlink. While such a configuration may not be generally appropriate, it is reasonable for this application. The PHY parameters used in the simulations are listed in Table C.2.

Table C.2: OFDM PHY Layer Parameters

Parameter	Value
Frequency (GHz)	3.5
Bandwidth (MHz)	20
Num. of subcarriers	200
Sampling factor	57/50
Cyclic prefix time (MHz)	1/4
Frequency sampling (MHz)	22.8
Useful symbol time (μs)	11.22
Symbol time (μs)	14.03

The topology used in the simulations is similar to the one depicted in figure C.1, and comprised of a number of SSs being served by a single BS; each SS accessed the video servers via the BS. The servers were connected to the BS through a 100 Mb/s interconnect with 1 μs propagation delay. It was assumed that the nodes were uniformly distributed in the area covered by the BS.

Using the methodology described in [161], the number of nodes that fell into the annulus corresponding to each modulation and coding scheme was determined. The maximum transmission range and the proportion of the entire coverage area for each modulation and coding scheme are given in Table C.3.

Five burst profiles were modeled in the simulations. Each had a dedicated buffer in the BS, as described in Section C.2.2. Since accumulation-based congestion control assumes sufficient buffer space so that no packets need to be dropped in the BS, a buffer size of 1000 KB was used in these simulations.

The video sources were selected from four excerpts of the mini-movie

Table C.3: Burst Profiles: Transmission Ranges and Coverage Areas

Burst Profile	*Receiver SNR* (dB)	*Trans. Range* (km)	*Surface* (%)
BPSK 1/2	6.4	11.54	39.40
QPSK 1/2	9.4	8.17	20.56
QPSK 3/4	11.2	6.64	27.95
16QAM 1/2	16.4	3.65	4.10
16QAM 3/4	18.2	2.97	5.15
64QAM 2/3	22.7	1.77	0.92
64QAM 3/4	24.4	1.45	1.92

"Standardized Evaluation Material" (StEM)[3]. This particular movie is available in 4K format (4096 by 1714 pixels, 24 frames/s), but was converted to CIF resolution (352 x 288) with a pixel aspect ratio of 16:9 for our experiments. This video quality was considered an appropriate balance between the quality requirements of the users and the capabilities of the wireless access network.

For encoding the videos the JSVM 5.2 reference software was used [175]. The encoded SVC bitstream contained one spatial layer with CIF resolution, and four temporal scalability levels enabling reconstruction at 24, 12, 6 and 3 frames per second. The AVC base layer was encoded using a hierarchical coding structure, a GOP size of 8 pictures, and an intra period of 24 pictures. The chosen GOP size was considered an appropriate balance between coding efficiency and delay in the congestion control loop (since adaptation is performed only once per GOP, i.e. every 1/3 of a second for the above case). In addition, one FGS layer was used to achieve SNR quality scalability. The four excerpts from StEM are listed in Table C.4.

As previously discussed, in the proposed adaptation policy, quality downscaling is performed before temporal downscaling. If the base layer pictures are predicted based on the enhancement layer reconstruction, then drift occurs in the prediction loop when parts of the FGS enhancement layers is removed [177]. Drift is prevented here by configuring the JSVM encoder to use only the quality base layer reconstruction for predicting other base layer pictures. This results in lower coding efficiency, and is clearly a design trade-off.

For these experiments, all streaming clients were configured to have an initial pre-buffering period of 1 second. This choice enables a fairly interactive end-user viewing experience, while being large enough to absorb considerable delay jitter

[3]This content is available under license from the Digital Cinema Initiative (DCI) and American Society of Cinematographers (ASC). Access procedure is available at www.dcimovies.com

Table C.4: Video Clips Used In The Simulation

Video Clip	*Length* (s)	*Mean Bitrate* (Kb/s)	*QP*	*PSNR* (dB)
STEM_A	53.1	525	38	36.47
STEM_B	53.1	497	38	36.13
STEM_C	53.1	510	35	38.70
STEM_D	53.1	508	32	41.37

caused by the base station buffer. The accumulation target a_i^* in Equation C.1 was initially set to 3000 bytes, roughly three times the MTU. This reflects [58], in which the authors use an accumulation target of three packets. Similarly, a value of $\kappa = 0.75$ was initially used in these experiments. Lastly, an MTU of 1024 bytes was chosen. Important parameter values are summarized in Table C.5 below.

Table C.5: Application and transport related simulation parameters

Parameter	*Value*
Playout buffer size (ms)	1000
Maximum Transmission Unit (Bytes)	1024
Accumulation Target a_i^* (Bytes)	3000
Congestion control parameter κ	0.75

C.3.2 Performance of the Adaptive Video Streaming System

To evaluate the performance of the proposed accumulation-based congestion control for adaptive streaming of H.264/SVC video, extensive simulations were performed for the simulation scenario presented in Section C.3.1. Table C.6 shows typical results from a simulation with twenty SSs uniformly distributed around the 802.16 BS.

Fairness

As fairness is an important concern for any distributed congestion control scheme, it is interesting to determine what level of fairness the proposed scheme delivers in this particular context. It is worth noting that the buffer mechanisms at the BS do lack some flexibility, which means that the congestion control mechanisms cannot deliver fair service to all subscriber stations in all situations.

Table C.6: Results from the simulation with 20 Subscriber Stations (SS).

SS	Start (s)	Video clip	Coding & Modulation	Th.put (Kb/s)	$\bar{a}_i$ (KB)	$\bar{rtt}_i$ (ms)	$\tilde{rtt}_{i,p}$ (ms)	TS (%)	PSNR (dB)
1	5.0	STEM_C	BPSK 1/2	404	1931	46.7	8.87	1.90	37.03
2	5.1	STEM_C	BPSK 1/2	407	1964	47.9	10.20	1.90	37.05
3	5.2	STEM_C	BPSK 1/2	407	1979	48.6	11.22	2.53	37.03
4	5.3	STEM_C	BPSK 1/2	401	1954	48.0	8.56	3.16	36.92
5	15.0	STEM_C	BPSK 1/2	413	1898	46.8	9.54	2.53	37.13
6	15.1	STEM_C	BPSK 1/2	403	1911	47.5	9.54	2.53	37.01
7	15.3	STEM_C	BPSK 1/2	406	1869	47.1	8.53	3.16	37.03
8	15.4	STEM_C	BPSK 1/2	408	1917	48.0	9.61	2.53	37.09
9	6.0	STEM_A	QPSK 1/2	367	1843	44.0	9.16	16.46	33.42
10	6.1	STEM_B	QPSK 1/2	383	1858	44.6	9.69	7.59	34.04
11	6.2	STEM_C	QPSK 1/2	409	1895	43.3	9.20	3.80	37.10
12	6.3	STEM_D	QPSK 1/2	367	1846	46.4	10.51	6.33	39.26
13	6.4	STEM_C	QPSK 3/4	383	2120	50.0	10.14	1.90	36.76
14	6.5	STEM_C	QPSK 3/4	381	2102	50.8	9.46	2.53	36.72
15	6.6	STEM_C	QPSK 3/4	387	2137	50.7	10.80	0.63	36.86
16	6.7	STEM_C	QPSK 3/4	382	2105	50.8	10.25	1.27	36.77
17	6.8	STEM_C	QPSK 3/4	379	2118	51.9	10.37	1.90	36.74
18	6.9	STEM_C	QPSK 3/4	383	2118	51.3	10.99	0.63	36.81
19	7.0	STEM_C	16QAM 1/2	373	2170	58.6	10.28	5.06	36.52
20	7.1	STEM_C	16QAM 3/4	372	2271	61.7	10.23	5.06	36.53

A simulation was performed in which some subscriber stations were activated early in the simulation and others were activated later. More specifically, 4 subscriber stations were activated 5 seconds after the start of the simulation and a further 4 SSs were activated 10 seconds later. The objective was to determine whether the congestion control mechanisms can quickly equalise (approximately) the amount of resources used by each subscriber station, thus exhibiting fairness.

Results are presented in table C.6. The results to focus on are the BPSK 1/2 results, as this is where the subscriber stations are added. The results show that there is very small variation in the data rates delivered to each SS, and that the video quality delivered to all clients is very much equal.

Figures C.3(b), C.3(c), and C.3(d) show the congestion window, accumulation and round-trip-time for two flows - that to the SS 2 client and that to the SS 5 client. The former was activated earlier in the simulation and the latter was activated later. The results show that the congestion control mechanism can quickly react to the arrival of new flows in the system and that the congestion windows and accumulation values for the two different flows reach approximate parity within a few seconds.

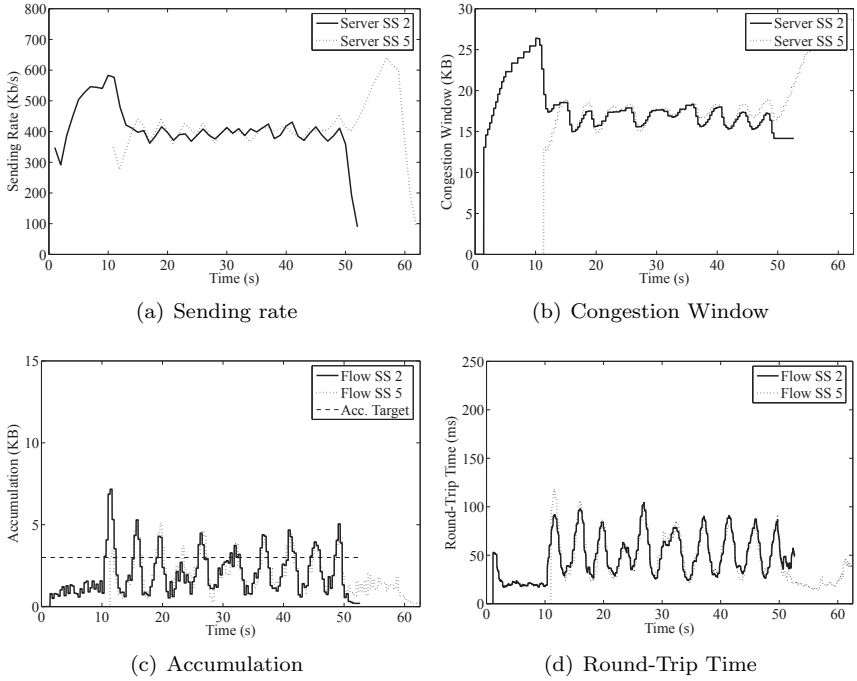
Finally, Jain's fairness metric [178] was calculated for all flows in the system. The resulting value of 0.9935 – compared to perfectly fair system with a metric of 1 – clearly demonstrates good fair sharing characteristics. Hence, it is clear that the system overall can quickly adapt to the arrival of new flows and quickly share the available resources between the users in a fair manner.

Utilization

Table C.7 shows different aspects of system utilization for the experiment described above. For each coding and modulation scheme, the average throughput on the physical and application layer is given, together with measures of system overhead as seen from the application layer, and average downlink utilization on the physical layer. With an overall system utilization of 96 %, it is clear that the adaptive mechanism results in efficient use of the available resources. Indeed, as some of the subscriber stations were activated a little after the simulation commenced, the maximal load is not offered to the system from the outset. If it were, the system utilization would have been even higher. This is evident from the lower utilization of the resources allocated to BPSK 1/2 users.

Sensitivity to congestion control parameters

To investigate the system's sensitivity to choice of congestion control parameters, a set of simulations were performed with different values of accumulation target a^* and congestion control constant κ. In the simulations, $\kappa = 0.5$ when a^* was varied. Similarly, $a^* = 3000$ when evaluating the impact of κ.



(a) Sending rate

(b) Congestion Window

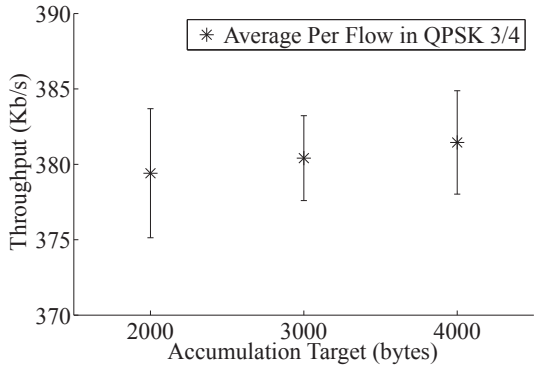
(c) Accumulation

(d) Round-Trip Time

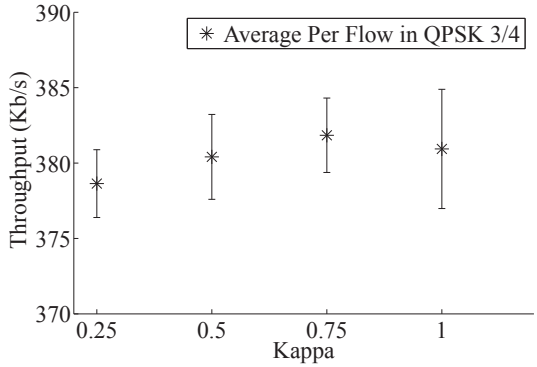
Figure C.3: Comparison of the two flows going to SS 2 and SS 5.

Figure C.4 shows how throughput varies for different values of a^* and κ for those flows that were using the QPSK 3/4 burst profile. The vertical error bars indicate the standard deviation of the throughput per flow, leading to an interpretation that smaller variance gives a higher fairness in terms of average throughput. The results show that throughput is insensitive to the values of these parameters for the simulation scenario investigated, with variations of under 1 %.

The a^* and κ parameters have a slightly higher impact in the case of the flows sharing the BPSK 1/2 resources, as can be seen from figure C.5. In that case, the mean throughput per flow increases by a few % with increases in both a^* and κ: Naturally, this results in a better overall system utilization. While the increase in throughput is not insignificant, it is not clear that substantial gains can be made from appropriate tuning of these parameters. The variation in the congestion window is also shown; from these figures, it can be seen that the larger value of κ results in slightly greater variation in the congestion window. Similarly, the peak round-trip times are higher when a^* is higher - this is due to the larger



(a) Throughput - a_i^*



(b) Throughput - κ

Figure C.4: Average throughput per flow for different values of κ and a_i^*.

amounts of data that can accumulate in the network. Further study is necessary to determine the impact of these parameters over a greater range of values and some different scenarios.

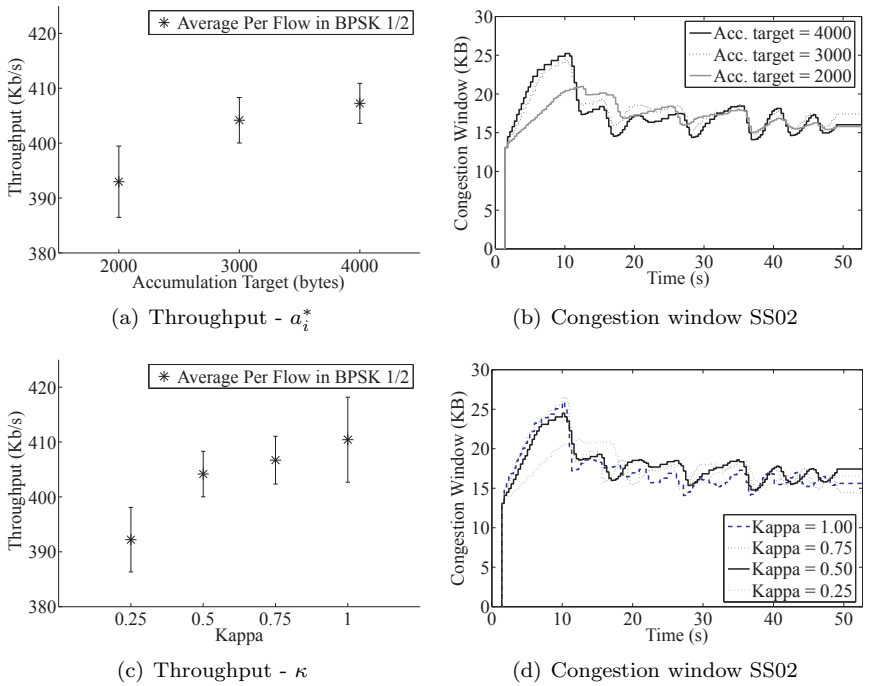
Streaming video quality

From the discussion in Section A.1.1, it is clear that assessing video quality objectively is a difficult task. This is particularly true in the case where the quality of the video varies with time, as happens here. Ultimately, subjective testing would have to be performed, and is an interesting topic for future work. However, the system implemented here does facilitate straightforward determination of one metric which has an impact on perceived video quality,

Table C.7: Utilization of the Wireless Link

Coding & Modulation	*Throughput* (Mb/s) *Max**	*PHY*	*APP***	*System Overhead*	*Downlink Utilization*
BPSK 1/2	3.562	3.234	3.197	10.25 %	90.8 %
QPSK 1/2	1.781	1.677	1.578	11.38 %	94.1 %
QPSK 3/4	2.672	2.616	2.411	9.76 %	97.9 %
16QAM 1/2	0.445	0.437	0.384	13.81 %	98.1 %
16QAM 3/4	0.445	0.437	0.383	13.99 %	98.1 %
Total:	8.906	8.401	7.953	11.84 %	96.0 %

* Theoretical throughput allocated to burst profile on the DL.
** Calculated in the period when all servers are transmitting.



(a) Throughput - a_i^*

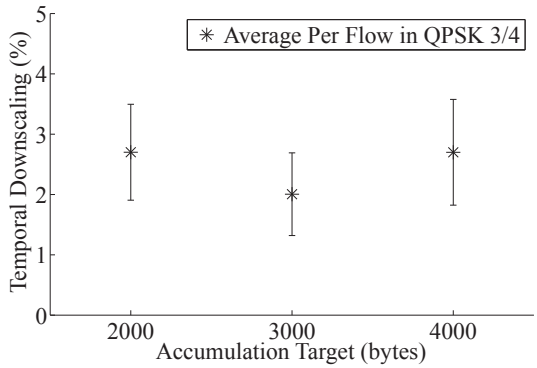
(b) Congestion window SS02

(c) Throughput - κ

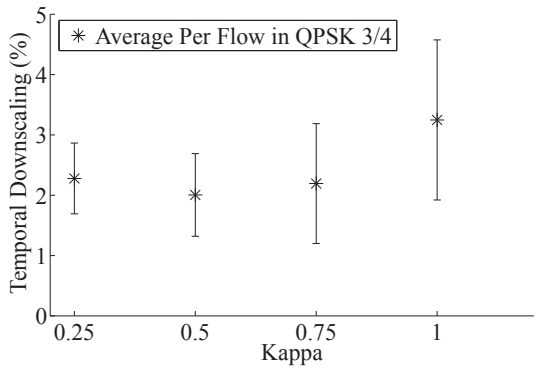
(d) Congestion window SS02

Figure C.5: Average throughput per flow, and congestion window as a function of time, for different values of κ and a_i^* in the BPSK 1/2 case

namely the amount of times the temporal scalability mechanisms is triggered



(a) Downscaling events - a_i^*



(b) Downscaling events - κ

Figure C.6: Frequency of temporal downscaling events for different κ and a_i^*.

during the streaming session. Since this leads to a change in temporal resolution, it obviously has more of an impact for certain types of video than for others. However, it is clear that such variations in general are undesirable, and consequently, it is discussed here as one important issue in the context of video quality.

Figure C.6 shows the frequency of occurrence of changes in temporal resolution for different values of a^* and κ. It can be seen that the frequency of occurrence of temporal downscaling is greater for larger values of κ. This is because this permits larger changes in transmission rate and, in particular, larger reductions in transmission rate when congestion is detected. On average, the temporal downscaling occurs every 15 s for this 24 frame/s test material.

Table C.8: Single Layer Video Clips Used In The Simulation

Video Clip	*Length* (s)	*Mean Bitrate* (Kb/s)	*QP*	*PSNR* (dB)
STEM_A_SL	53.1	364	34	35.23
STEM_B_SL	53.1	391	32	36.05
STEM_C_SL	53.1	353	30	38.02
STEM_D_SL	53.1	370	27	40.63

Table C.9: Utilization of the Wireless Link for Single Layer AVC

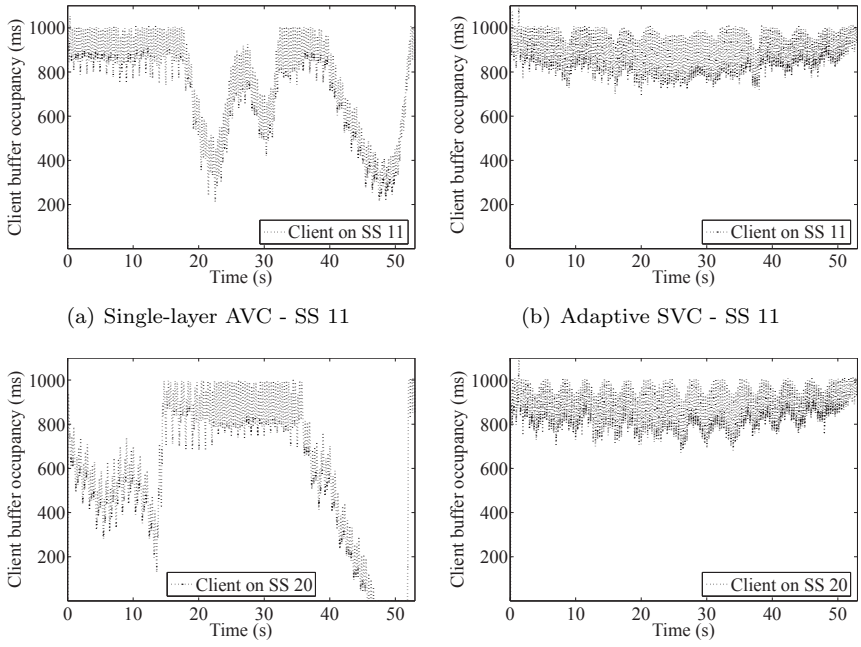
Coding & Modulation	*Throughput* (Mb/s) *Max**	*PHY*	*APP***	*System Overhead*	*Downlink Utilization*
BPSK 1/2	3.562	2.800	2.866	19.56 %	78.6 %
QPSK 1/2	1.781	1.601	1.552	12.89 %	89.9 %
QPSK 3/4	2.672	2.422	2.137	20.01 %	90.6 %
16QAM 1/2	0.445	0.418	0.354	20.43 %	93.9 %
16QAM 3/4	0.445	0.419	0.354	20.43 %	94.1 %
Total:	8.906	7.659	7.263	18.66 %	89.4 %

* Theoretical throughput allocated to burst profile on the DL.
** Calculated in the period when all servers are transmitting.

Lastly, in order to illustrate the visual quality at the receiver side, and the perceptual effects of reducing the frame rate, some of the reconstructed video sequences are placed on the author's web page at [179].

C.3.3 Comparison with single-layer H.264/AVC

In order to compare our approach with a single-layer streaming solution, the JSVM encoder was configured to use only one spatial and quality layer. The temporal prediction structure was identical to the scalable case, with a GOP size of eight pictures. Further, the quantization parameter was adjusted so that the resulting average bitrate was similar to the throughput obtained in the simulations in chapter C.3.2. Details of the single layer H.264/AVC video clips used in these simulations are given in Table C.8. Table C.9 summarizes the performance of the 802.16 network for a sample simulation. Comparing with Table C.7, it is evident that the utilization of system resources is significantly lower. Simulations were also performed in which clip STEM_C_SL was encoded at a larger average rate of 391 Kb/s (QP=29), but then all the clients in the system experienced playout buffer underflow and interruptions in continuous playback.



(a) Single-layer AVC - SS 11

(b) Adaptive SVC - SS 11

(c) Single-layer AVC - SS 20

(d) Adaptive SVC - SS 20

Figure C.7: Comparison of buffer occupancies for single-layer H.264/AVC and adaptive H.264/SVC-based video streaming solutions.

For the chosen single-layer simulation scenario, the streaming clients on subscriber stations 13 to 20 all experience playout buffer underflow events. This is in stark contrast to the SVC case, in which no subscriber station experienced buffer underflow.

Figure C.7 shows the client playout buffer occupancies over time for SS 11 and 20, for both the single-layer AVC case and the proposed scalable SVC streaming solution. The short-term fluctuation in buffer occupancy is due to the scheduling being performed on a GOP-by-GOP basis on the server side – packet transmission times are distributed so that the sending rate is as smooth as possible. In time, one GOP corresponds to the duration of 8 frames (1/3 s). From the plots in figure C.7, one can clearly see the advantage and effectiveness of the proposed adaptive streaming solution. As long as the client playout buffer is large enough to absorb inevitable delay jitter and the extra delay caused by the packet scheduling mechanism, a high and relatively stable buffer occupancy can be maintained at the receiving client.

C.4 Results using the Streaming Media Testbed

This section presents an evaluation of the adaptive streaming video system described in Section C.2.3 above, using the streaming media testbed described in Part B. More specifically, network emulation was employed to show how the rate-adaptation mechanisms perform over a fixed bandwidth channel, and to investigate the sensitivity of the system to delay on the forward and reverse path.

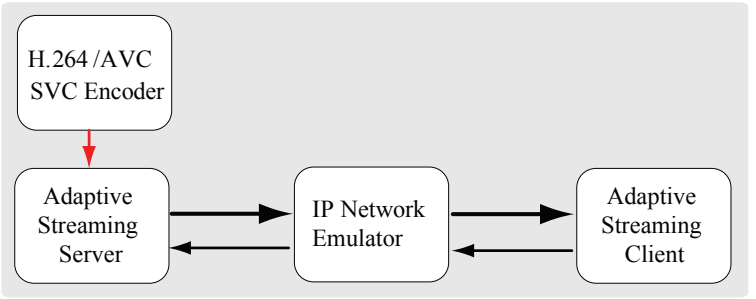


Figure C.8: Setup for network emulation experiments

C.4.1 Experimental Setup

Figure C.8 shows the setup used in these experiments. The streaming server and client were interconnected on the test network, and the IP network emulator was used to add a varying amount of delay on the link, and to restrict the link bandwidth. The emulator that was used supports three different delay impairments; a certain constant latency, or random latency following either a uniform or a normal/gaussian distribution. Studies from literature indicate that delays in the Internet can usually be characterized by an asymmetric unimodal distribution with a long tail of higher positive values [180]. Unfortunately, no such distribution was supported by the network emulator. For these experiments, the normal distribution was used with varying average latency, and with a fixed variance of 2 ms. The video clip used in the experiments was STEM_C (see Table C.4 in Section C.3).

C.4.2 Fixed Bandwidth Network Link

From Figure C.9, it is clear that the streaming server is capable of effectively adapting the video transmission rate to the available bandwidth. Further, the application throughput is very smooth, even though some variations can be seen

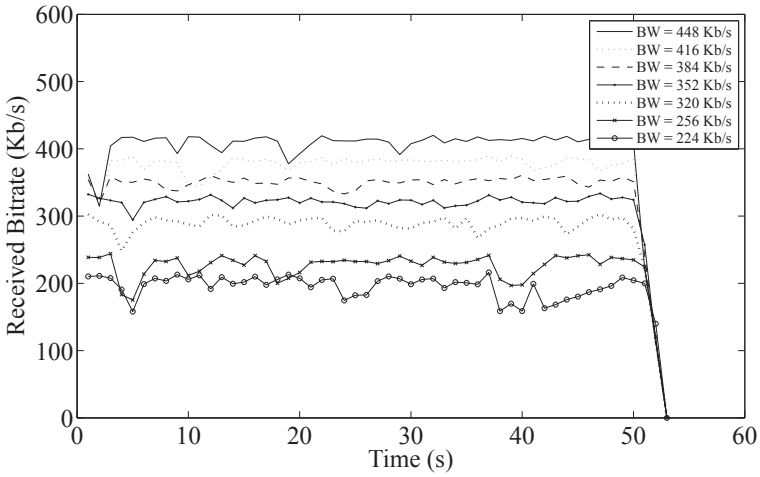


Figure C.9: Received bitrate for experiments with different link bandwidth.

in the graphs. In the case of 224 Kb/s available bandwidth, for one third of the GOPs, only the base representation of the base layer I/P pictures is transmitted. This indicates a situation in which the bitrate target was exceeded, thereby causing more severe congestion. Playout buffer underflows were not experienced in any of the experiments.

When the available bandwidth is high compared to the maximum bitrate of the video, the scheme has a more difficult time utilizing all of the available bandwidth. The drops in received bitrate should be attributed to the fact that – for some GOPs – the maximum bitrate of all scalable layers is below the target bitrate. Also, as described in Section C.2.3, remember that if the amount of video data is less then the target bitrate for a GOP, the congestion window is not adjusted upwards.

C.4.3 Delay on the Forward Path

Figure C.10 shows the received bitrate at the client for experiments in which average latency of 5, 20, 50, 75 and 100 ms are added on the forward link. In addition, delay and delay jitter is incurred by the bandwidth restriction, which was set to 352 Kb/s. As can be seen, up to 75 ms of additional delay does not have any significant impact on the performance of the adaptive scheme. For delays of 100 ms, the overestimation of accumulation – as depicted in Figure C.11 – clearly has a severe impact on application performance.

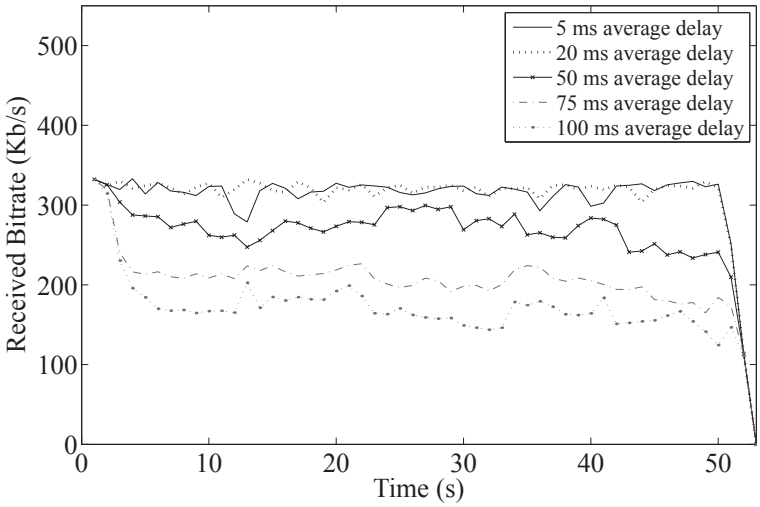
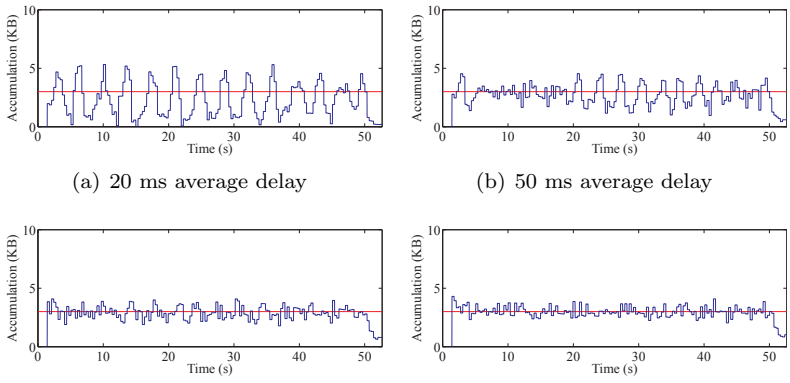


Figure C.10: Received bitrate for experiments with different average delay on the forward link.



(a) 20 ms average delay

(b) 50 ms average delay

(c) 75 ms average delay

(d) 100 ms average delay

Figure C.11: Accumulation estimated at the server for increasing delays on forward path. Link bandwidth is additionally restricted to 352 Kb/s

C.4.4 Delay on the Reverse Path

To investigate the system sensitivity to reverse-path congestion, which is a common problem to accumulation-based protocols (see section 2.2.3), experiments were also performed in which latency from 5 ms to 100 ms was added on the

reverse feedback path. The link bandwidth was constrained in both directions, more specifically 352 Kb/s on the forward path, and 64 Kb/s on the reverse path. The results in Figure C.12 indicate that the application throughput is not significantly affected by delays of less than 75 ms on the reverse feedback channel.

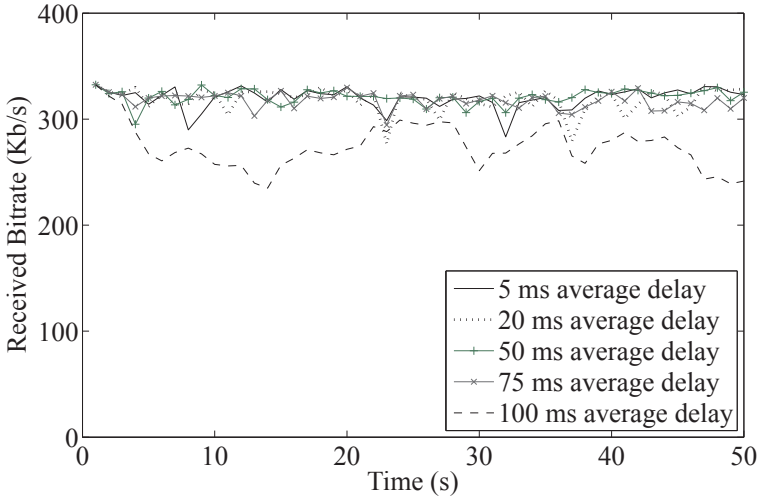


Figure C.12: Received bitrate for experiments in which average delays of 5, 20, 50, 75 and 100 ms are added on the reverse feedback path.

C.5 Summary and Discussion

This chapter has presented an adaptive video-on-demand streaming solution for IEEE 802.16 broadband wireless networks. The proposed architecture is based on H.264/MPEG-4 Scalable Video Coding for temporal and fine-granular quality adaptation during a streaming session, together with an accumulation-based congestion control scheme that, unlike mechanisms such as TFRC, does not rely on packet loss to perform efficient congestion control.

The system was shown to deliver high utilization of the wireless link, and a fair sharing of network resources while maintaining smooth variations in throughput for the video applications. The scheme also ensures that no interruptions in continuous video playback occur during the streaming session.

Simulation results provided in this chapter suggest that 96 % of the resources in an IEEE 802.16 wireless network can be efficiently utilized by network-adaptive video applications. Clearly, the number of users that can be served by such

a system is dependent on many factors, such as the bitrate requirements and characteristics of the video applications, together with the size of the FBWA coverage area and spatial distribution of users within that coverage area. With a realistic uniform spatial distribution of SS and a focus on rural area deployment, it was shown how twenty simultaneous users could receive on-demand video services at just below 400 Kb/s on average. Thus, the system provided around 8 Mb/s of total bandwidth to the video applications.

For the chosen scenario, video clips, and video adaptation policy - the temporal downscaling mechanism was used on average 2-3 % of the time, and a pre-buffering period of 1 second was more than enough to keep the clients from experiencing buffer underflow. Some analysis of the sensitivity of the system to the algorithm parameters was performed: the results showed that for the scenario investigated, the parameters did not have a very significant impact on the overall performance. Lastly, a comparison between an adaptive SVC-based solution and a non-adaptive H.264/AVC solution was performed: the SVC-based solution clearly performed much better, providing higher utilization or more efficient use of the wireless system resources. Further, it is important to note that no clients ever suffered buffer underflow and interruption in playback for the adaptive streaming system, in contrast to the non-adaptive single-layer AVC case. Thus, the proposed SVC-based solution for network-adaptive streaming video may provide end-users with an improved and enhanced experience of streaming media services.

Finally, an evaluation of the rate-adaptive streaming video system was also carried out using network emulation and the testbed described in Part B. The results show that the adaptive scheme effectively adapts the transmission rate to the available bandwidth, and indicated that the system is fairly insensitive to gaussian distributed delays below 75 ms (on average and with a variance of 2 ms) on the forward path or the reverse feedback path.

Future work could involve investigating the impact of the algorithm parameters for a wider range of values and for more scenarios. Also, it is interesting to explore how the system operates if the resource allocation in the BS is made more dynamic. In addition, it would be very valuable to further investigate the perceptual impact of reducing the frame rate for the proposed scheme. Several aspects of the accumulation-based congestion control deserves more attention, e.g. regarding the parameter sensitivity to a greater range of values and different scenarios, and the effect of the oscillatory behavior seen in the plots depicting round-trip-time and accumulation.

Chapter 3

Conclusion

This thesis considers some aspects of multimedia communication over unreliable and resource-constrained IP-based packet-switched networks. The focus and objectives of the work is related to estimating, evaluating and enhancing the quality of streaming video services and applications.

Towards this end, the first part of the thesis focused on measuring video quality in a video communication system at the receiver side. Specifically, a low-complexity method for estimating block-edge impairments in video has been developed, and the performance of a quality metric measuring the strengths of such impairments is evaluated. Since the method does not depend on the availability of a reference video signal, it may be applied at the receiving end of a video communication system. Results show that the performance of the metric was best for intra-coded frames, since block-edge impairments are most prominent in such frames (for the coding scheme that was investigated). To be able to effectively assess video quality for other frame types and new coding schemes like H.264/AVC, other types of impairments (considering e.g. blurriness) need to be integrated into the quality metric.

The next part of the thesis considers how the performance of streaming media application could be evaluated in a controlled manner. Part B presents a laboratory testbed for this purpose, which consists of media applications, network devices, sophisticated traffic monitoring and capture tools, together with a device that enables accurate regeneration of media traffic flows. This represents an important step towards realistic and reproducible experimental testing of streaming media applications. For instance, once subjected to a specific type of network impairment (e.g. loss and delay), a media flow can be accurately replayed to a receiving streaming media client. The performance of the packet flow regenerator has been investigated, and results show how accurately a video

flow could be reproduced; e.g. in the range of 5-20 Mb/s, over 99 % of the packet inter-arrival times are within 2 ms of the intended value.

Illustrating the usefulness of the testbed described above, the error robustness of a high-definition H.264/AVC broadcast service has been evaluated. A broadcast video application was subjected to uniformly distributed packet loss using a real-time network emulation device, showing the effectiveness of some error resilience tools in the H.264/AVC standard such as slice partitioning and flexible macroblock ordering. In order to use the reference decoder in this evaluation, which is not optimized for real-time operation (especially not for high-definition video applications), an application has been developed that in an offline fashion assembles the received video stream from the traffic captured at the output of the network emulator.

The last part of the thesis is an application and system study, focusing on how the the performance of an adaptive video streaming can be improved and enhanced by utilizing a novel media-friendly rate-adaptation algorithm. Specifically, an adaptive video-on-demand solution based on H.264/AVC scalable video coding is presented, which is shown to enable efficient video distribution over IEEE 802.16 fixed broadband wireless networks. The proposed congestion and rate control is based on ideas from accumulation-based congestion control, and allows streaming media applications to have a smoother transmission rate, while effectively reacting to congestion in the network in a timely manner. A prototype of the video streaming system has been developed and presented, and the overall system performance is evaluated using a simulated 802.16 network model, and the testbed described in Part B.

Future Work

The development of NR quality metrics that estimate video quality at the receiver side remain an important research goal, and will be essential in developing fully automated quality monitoring and assessment systems for multimedia services and applications.

The proposed NR metric quantifies the effect of block-edge impairments, which are most relevant for MPEG-2 and early MPEG-4 video systems. With the standardization and adoption of H.264/MPEG-4 AVC (and in the future maybe also SVC), methods for assessment of video quality at the receiver side must take new types of visual impairments into account. For H.264/MPEG-4 AVC, block-edge impairments are no longer the dominant compression-related distortion, and methods for determining the impact of blurriness and ringing will become more important. Further, evaluating the impact of transmission-related distortions such as packet loss for these new compression schemes in an automated manner, is a difficult problem which is not yet fully solved. Finally,

both AVC and SVC support temporal scalability, so methods for automatically evaluating the perceived impact of varying the frame rate for different types of video content will be essential. With the adoption of SVC, this may also apply to other types of video adaptation techniques.

The development and validation of new NR metrics require formal subjective testing in addition to comparison with state-of-the-art objective FR video quality models. In order to evaluate and predict the perceptual effects of compression, transmission and adaptation-related distortions, highly realistic delivery scenarios and appropriate processed test material must be used as starting point for subjective testing and model development. Therefore, continued efforts are required for developing more realistic simulation scenarios, and for improving the realism and repeatability of experimental media delivery testbeds. Only in this way can our studies better reflect real-life conditions.

With regard to the adaptive streaming solution presented in this thesis, future work could include studying other scenarios, and further investigating the impact of varying system parameters. In addition, it would be interesting to explore the effect that dynamic resource allocation mechanisms have on the performance of the streaming video system. Finally, several aspects of the accumulation-based congestion control deserve more attention, e.g. regarding parameter sensitivity, and the effect of the oscillatory behavior seen in the plots for round-trip time, accumulation and transmission rate.

A permanent record of all code used and developed in this work will be archived at the Centre for Quantifiable Quality of Service in Communication Systems (NTNU) for future reference.

References

[1] Apple Inc., "iPod," http://www.apple.com, 2007.

[2] YouTube Inc., "Broadcast yourself," http://www.youtube.com, 2006.

[3] A. Perkis, Y. Abdeljaoued, C. Christopoulos, T. Ebrahimi, and J. F. Chicharo, "Universal Multimedia Access from Wired and Wireless Systems," *Circuits, Systems, and Signal Processing; Special issue on Multimedia Communications*, vol. 20, no. 3, pp. 387–402, 2001.

[4] M. S. Blumenthal and D. D. Clark, "Rethinking the Design of the Internet: the End-to-end Arguments vs. the Brave New World," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 70–109, 2001.

[5] F. Pereira and I. Burnett, "Universal multimedia experiences for tomorrow," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 63–73, 2003.

[6] S. Floyd and V. Paxson, "Difficulties in simulating the internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, 2001.

[7] T. Wiegand, G. J. Sullivan, J.-R. Ohm, and A. Luthra, "Meeting Report, Draft 6," JVT-V200 (Output Document (Draft) of Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), Jan. 2007. [Online]. Available: http://ftp3.itu.ch/av-arch/jvt-site/

[8] F. C. Pereira and T. Ebrahimi, *The MPEG-4 Book*. Prentice Hall PTR, 2002.

[9] *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*, 3rd ed. ISO/IEC 14496-10:2005, 2005.

[10] *Advanced Video Coding for Generic Audiovisual Services*, 03/2005 ed. ITU-T H.264, 2005.

[11] T. Wiegand, G. J. Sullivan, G. Bjontegaard, 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, 2003.

[12] G. J. Sullivan and T. Wiegand, "Video Compression - from Concepts to the H.264/AVC Standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[13] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG-4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[14] *Advanced Video Coding for Generic Audiovisual Services*. ITU-T Rec. H.264 and ISO/IEC 14496-10 MPEG-4 part 10 Advanced Video Coding (AVC), 2003.

[15] 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, 2003.

[16] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," JVT-PO14 (Input Document to Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), July 2005. [Online]. Available: http://ftp3.itu.ch/av-arch/jvt-site/

[17] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with h.264/avc: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[18] M. M. Hannuksela, W. Ye-Kui, and M. Gabbouj, "Isolated regions in video coding," *Multimedia, IEEE Transactions on*, vol. 6, no. 2, pp. 259–267, 2004.

[19] T. Halbach and S. Olsen, "Error Robustness Evaluation of H.264/MPEG-4 AVC," in *Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP'04)*, S. Panchanathan and B. Vasudev, Eds., vol. 3508, Jan. 2004, pp. 617–627.

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

[21] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

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

[23] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Extension of the H.264/MPEG-4 AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology, to appear*, p. Draft, Summer 2007.

[24] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. e. Wien, "Joint Draft of SVC Amendment," JVT-U202 (Output Document from Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), Oct. 2006. [Online]. Available: http://ftp3.itu.ch/av-arch/jvt-site/

[25] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model JSVM-5," JVT-R202 (Output Document from Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), Jan. 2006. [Online]. Available: http://ftp3.itu.ch/av-arch/jvt-site/

[26] Y.-K. Wang, M. M. Hannuksela, S. Pateux, and A. Eleftheriadis, "System and transport interface of the emerging svc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, p. Draft, 2007.

[27] 3rd Generation Partnership Project (3GPP), "Technical Spesification TS 26.234, Transparent end-to-end Packet-switched Streaming Service (PPS); Protocols and codecs (Release 7)," Sep. 2006.

[28] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: approaches and directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282–300, 2001.

[29] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[30] A. Perkis, P. Svensson, O. I. Hillestad, S. Johansen, J. Zhang, A. Sæbø, and O. Jetlund, "Multimedia over IP Networks," *Telektronikk*, vol. 1, no. Real-time communication over IP, pp. 43–53, 2006.

[31] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 3550 (Standard), July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[32] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," RFC 4340 (Proposed Standard), Mar. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4340.txt

[33] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP payload format for H.264 video," RFC 3984 (Proposed Standard), Feb. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3984.txt

[34] T. Friedman, R. Caceres, and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)," RFC 3611 (Proposed Standard), Nov. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3611.txt

[35] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326 (Proposed Standard), Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[36] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," RFC 2543 (Proposed Standard), Mar. 1999, obsoleted by RFCs 3261, 3262, 3263, 3264, 3265. [Online]. Available: http://www.ietf.org/rfc/rfc2543.txt

[37] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC 2327 (Proposed Standard), Apr. 1998, obsoleted by RFC 4566, updated by RFC 3266. [Online]. Available: http://www.ietf.org/rfc/rfc2327.txt

[38] S. Wenger, Y.-K. Wang, and T. Schierl, "RTP Payload Format for SVC Video," Internet Draft, Oct. 2006. [Online]. Available: http://tools.ietf.org/wg/avt/draft-wenger-avt-rtp-svc-03.txt

[39] ——, "Transport and Signaling of SVC in IP Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, p. Draft, 2007.

[40] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.

[41] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. Stockholm, Sweden: ACM Press, 2000, pp. 43–56.

[42] J. Vieron and C. Guillemot, "Real-time constrained tcp-compatible rate control for video over the internet," *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 634–646, 2004.

[43] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and tcp-friendly congestion control for scalable video streams," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 196–206, 2006.

[44] M. Welzl and W. Stadler, "User-centric evaluation of tcp-friendly congestion control for real-time video transmission," *Elektrotechnik und Informationstechnik*, no. June, 2005.

[45] M. Vojnovic and J. Y. Le Boudec, "On the long-run behavior of equation-based rate control," vol. 13, no. 3, pp. 568–581, 2005.

[46] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 339–355, 2001.

[47] D. Miras and G. Knight, "Smooth quality streaming of live internet video," in *IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 2, 2004, pp. 627–633 Vol.2.

[48] D. Loguinov and H. Radha, "End-to-end rate-based congestion control: convergence properties and scalability analysis," vol. 11, no. 4, pp. 564–577, 2003.

[49] S.-j. Bae and S. Chong, "Tcp-friendly flow control of wireless multimedia using ecn marking," *Signal Processing: Image Communication*, vol. 19, no. 5, pp. 405–419, 2004.

[50] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703–717, 2003.

[51] M. Chen and A. Zakhor, "Multiple tfrc connections based rate control for wireless networks," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1045–1062, 2006.

[52] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): protocol specification," RFC 3448 (Proposed Standard), Jan. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3448.txt

[53] B. Briscoe, "Flow Rate Fairness: Dismantling a Religion," Internet Draft, Oct. 2006. [Online]. Available: http://www.ietf.org/internet-drafts/draft-briscoe-tsvarea-fair-00.txt

[54] S. Floyd, "Metrics for the Evaluation of Congestion Control Mechanisms," Internet Draft, 2006. [Online]. Available: http://www.ietf.org/internet-drafts/draft-irtf-tmrg-metrics-06.txt

[55] L. Massoulie and J. Roberts, "Bandwidth sharing: objectives and algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 320–328, 2002.

[56] F. Kelly, A. Maolloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[57] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[58] Y. Xia, D. Harrison, S. Kalyanaraman, K. Ramachandran, and A. Venkatesan, "Accumulation-based Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 69–80, 2005.

[59] M. Etoh and T. Yoshimura, "Advances in Wireless Video Delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 111–122, 2005.

[60] P. Bucciol, G. Davini, E. Masala, E. Filippi, and J. C. De Martin, "Cross-layer Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless Networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, 2004, pp. 3027–3031 Vol.5.

[61] I. V. Bajic, "Noncausal error control for video streaming over wireless packet networks," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1263–1273, 2006.

[62] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low-delay video streaming over error-prone channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 841–851, 2004.

[63] A. S. Bopardikar, O. I. Hillestad, and A. Perkis, "Temporal concealment of packet loss related distortions in video based on structural alignment," in *Eurescom Summit 2005*, Heidelberg, Germany, April 27–29 2005.

[64] F. Verdicchio, A. Munteanu, A. I. Gavrilescu, J. Cornelis, and P. Schelkens, "Embedded multiple description coding of video," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3114–3130, 2006.

[65] S. Rane, A. Aaron, and B. Girod, "Error-resilient video transmission using multiple embedded wyner-ziv descriptions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, 2005, pp. II–666–9.

[66] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan, "Forward error correction (fec) codes based multiple description coding for internet video streaming and multicast," *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 745–762, 2001.

[67] F. Tao and C. Lap-Pui, "Gop-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1323–1330, 2006.

[68] Y. Liu and S. Yu, "Adaptive unequal loss protection for scalable video streaming over ip networks," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1277–1282, 2005.

[69] G. Tong, G. Lu, and M. Kai-Kuang, "Reducing video-quality fluctuations for streaming scalable video using unequal error protection, retransmission, and interleaving," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 819–832, 2006.

[70] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Channel-adaptive resource allocation for scalable video transmission over 3g wireless network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 8, pp. 1049–1063, 2004.

[71] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming over internet packet traces," in *IEEE International Conference on Image Processing (ICIP 2005)*, vol. 2, 2005, pp. 161–164.

[72] P. A. Chou and M. Zhourong, "Rate-distortion optimized streaming of packetized media," *Multimedia, IEEE Transactions on*, vol. 8, no. 2, pp. 390–404, 2006.

[73] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2003, pp. III–661–4 vol.2.

[74] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2004, pp. 179–182.

[75] J. Chakareski and P. Frossard, "Rate-Distortion Optimized Packet Scheduling Over Bottleneck Links," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2005, pp. 1066–1069.

[76] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," RFC 2475 (Informational), Dec. 1998, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[77] Z. Fan, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "A novel cost-distortion optimization framework for video streaming over differentiated services networks," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2003, pp. III–293–6 vol.2.

[78] Y. Andreopoulos, R. Keralapura, M. van der Schaar, and C. N. Chuah, "Failure-aware, open-loop, adaptive video streaming with packet-level optimized redundancy," *Multimedia, IEEE Transactions on*, vol. 8, no. 6, pp. 1274–1290, 2006.

[79] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, 2001.

[80] Y.-G. Kim, J. Kim, and C. C. J. Kuo, "Tcp-friendly internet video with smooth and fast rate adaptation and network-aware error control."

[81] T. Kim and M. H. Ammar, "Optimal quality adaptation for scalable encoded video," vol. 23, no. 2, pp. 344–356, 2005.

[82] D. Mukherjee, E. Delfosse, K. Jae-Gon, and W. Yong, "Optimal Adaptation Decision-taking for Terminal and Network Quality-of-Service," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 454–462, 2005.

[83] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for Video Delivery over Wireless Internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123–134, 2005.

[84] G. M. Muntean, P. Perry, and L. Murphy, "A new adaptive multimedia streaming system for all-ip multi-service networks," *Broadcasting, IEEE Transactions on*, vol. 50, no. 1, pp. 1–10, 2004.

[85] N. Cranley, "User-Perceived Quality-Aware Adaptation of Streamed Multimedia over Best-effort IP Networks," PhD Thesis, University College Dublin, 2004.

[86] S. Feng, G.-h. Er, Q.-h. Dai, and Y.-b. Liu, "An optimal quality adaptation mechanism for end-to-end fgs video fgs video transmission," *Journal of Zhejiang University - Science AProceedings of the International Packet Video Workshop*, vol. 7, no. 0, pp. 119–124, 2006.

[87] ITU-R Recommendation BT.500-11, "Methodology for the Subjective Assessment of the Quality of Television Pictures," Nov. 2002.

[88] ITU-T Recommendation P.910, "Subjective Video Quality Assessment Methods for Multimedia Applications," Sep. 1999.

[89] ITU-T Recommendation P.911, "Subjective Audiovisual Quality Assessment Methods for Multimedia Applications," Dec. 1998.

[90] L. Lu, Z. Wang, A. C. Bovik, and J. Kouloheris, "Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of mpeg video," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2002, pp. 61–64 vol.1.

[91] S. Winkler and A. Sharma and D. McNally, "Perceptual video quality and blockiness metrics for multimedia streaming applications," in *Proc. 4th International Symposium on Wireless Personal Multimedia Communications*, Aalborg, Denmark, September 2001, pp. 553–556.

[92] S. Winkler, *Digital Video Quality – Vision Models and Metrics*. John Wiley & Sons, Jan. 2005.

[93] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[94] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[95] H. R. Sheikh and A. C. Bovik, "Image Information and Visual Quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[96] A. B. Watson, J. Hu, and J. F. McGowan, "Digital Video Quality Metric based on Human Vision," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 20–29, 2000.

[97] M. A. Masry and S. S. Hemami, "A Metric for Continuous Quality Evaluation of Compressed Video with Severe Distortions," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 133–146, 2004.

[98] H. R. Wu, M. Yuen, and B. Qiu, "Video coding distortion classification and quantitative impairment metrics," in *Proceedings of the 3rd International Conference on Signal Processing*, vol. 2, 1996, pp. 962–965.

[99] M. Yuen and H. R. Wu, "A survey of hybrid MC/DPCM/DCT video coding distortions," *Signal Processing*, vol. 70, no. 3, pp. 247–278, Nov. 1998.

[100] J. Caviedes and F. Oberti, "A new sharpness metric based on local kurtosis, edge and energy information," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 147–161, 2004.

[101] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "Perceptual Blur and Ringing Metrics: Application to JPEG2000," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 163–172, 2004.

[102] H. R. Wu and M. Yuen, "A generalized block-edge impairment metric for video coding," *IEEE Signal Processing Letters*, vol. 4, no. 11, pp. 317–320, 1997.

[103] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1, 2002, pp. 477–480.

[104] W. Gao, C. Mermer, and Y. Kim, "A De-blocking Algorithm and a Blockiness Metric for Highly Compressed Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1150–1159, 2002.

[105] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3440–3451, 2006.

[106] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *Multimedia, IEEE Transactions on*, vol. 6, no. 2, pp. 327–334, 2004.

[107] D. Patel and L. F. Turner, "Effects of ATM Network Impairments on Audio-visual Broadcast Applications," in *Proceedings of the IEE Vision, Image and Signal Processing*, vol. 147, no. 5, Oct. 2000.

[108] S. Vorren, "Subjective Quality Evaluation of the Effect of Packet Loss in High-Definition Video," M.Sc. Thesis, Norwegian University of Science and Technology, 2006.

[109] VQEG (Video Quality Experts Group), "Multimedia group test plan draft v1.11." [Online]. Available: www.vqeg.org, Feb. 2006.

[110] ——, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II," [Online]. Available: www.vqeg.org, August 2003.

[111] O. I. Hillestad, V. Radhakrishnan, A. S. Bopardikar, and A. Perkis, "Video Quality Evaluation for UMA," in *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (Wiamis 2004)*, Lisboa, Portugal, April 21–23 2004.

[112] V. Radhakrishnan, A. S. Bopardikar, A. Perkis, and O. I. Hillestad, "No-reference metrics for video streaming applications," in *International Packet Video Workshop*, Irvine, USA, December 13–14 2004.

[113] XVID, "MPEG-4 ASP Codec," http://www.xvid.org/, 2006.

[114] ns-2, "Network Simulator," www.isi.edu/nsnam/ns/, 2006.

[115] Emulab, "Network Emulation Testbed Home," http://www.emulab.net/, 2006.

[116] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*. Boston, MA: USENIX Association, Dec. 2002, pp. 255–270.

[117] C.-N. Wang, C.-Y. Tsai, H.-C. Chuang, Y.-C. Lin, J.-H. Chen, K. L. Tong, F.-C. Chang, C.-J. Tsai, S.-Y. Lee, T. Chiang, and H.-M. Hang, "Fgs-based video streaming test-bed for mpeg 21 universal multimedia access with digital item adaptation," in *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS '03)*, vol. 2, 2003, pp. II–364–II–367 vol.2.

[118] E. Setton, A. Shionozaki, and B. Girod, "Real-time streaming of prestored multiple description video with restart," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, 2004, pp. 1323–1326 Vol.2.

[119] R. J. Green, S. I. Woolley, N. W. Garnham, and K. P. Jones, "Experimental testbed results for broadband residential video service qos management," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 2, 2002, pp. 1142–1148 vol.2.

[120] W. Kellerer, E. Steinbach, P. Eisert, and B. Girod, "A real-time internet streaming media testbed," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, 2002, pp. 453–456 vol.2.

[121] C. Emmanuel and G. Johan, "Increasing the determinism of network emulation to evaluate communication protocols," in *Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology*. Toulouse, France: ACM Press, 2005.

[122] J. Garcia, F. Alfredsson, and A. Brunstrøm, "The impact of loss generation on emulation-based protocol evaluation," in *Proceedings of the International Conference on Parallel and Distributed Computing and Networks (PDCN 2006)*, Innsbruck, Austria, 2006.

[123] N. Spring, L. Peterson, A. Bavier, and V. Pai, "Using planetlab for network research: myths, realities, and best practices," *SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 17–24, 2006.

[124] L. Peterson and T. Roscoe, "The design principles of planetlab," *SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 11–16, 2006.

[125] N. Brownlee and K. C. Claffy, "Understanding Internet Traffic Streams: Dragonflies and Tortoises," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 110–117, 2002.

[126] J. W. Roberts, "Traffic theory and the internet," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 94–99, 2001.

[127] H. Pucha, Y. C. Hu, and Z. M. Mao, "On the impact of research network based testbeds on wide-area experiments," in *6th ACM SIGCOMM on Internet measurement*. Rio de Janeriro, Brazil: ACM Press, 2006, pp. 133–146.

[128] O. I. Hillestad, B. Libak, and A. Perkis, "Performance Evaluation of Multimedia Services Over IP Networks," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Amsterdam, The Netherlands, July 6–8 2005, pp. 1464–1467.

[129] O. I. Hillestad, O. Jetlund, and A. Perkis, "Error robustness evaluation of high quality h.264/avc broadcast services over rtp/ip using network emulation," Norsk Nettforskningsseminar, October 27–28 2005.

[130] ——, "RTP-based Broadcast Streaming of High Definition H.264/AVC Video: An Error Robustness Evaluation," *Journal of Zhejiang University Presented at theInternational Packet Video Workshop*, vol. 7, no. 0, pp. 19–26, 2006.

[131] Uninett, "The Norwegian Research Network," http://www.uninett.no, 2007.

[132] C. Perkins, *RTP - Audio and Video for the Internet*. Addison-Wesley Professional, June 2003.

[133] Envivio, "IP-based MPEG-4 solutions," www.envivio.com, 2006.

[134] J. Chakareski, J. Apostolopoulos, S. Wee, T. Wai-tian, and B. Girod, "R-D hint tracks for low-complexity R-D optimized video streaming," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, 2004, pp. 1387–1390 Vol.2.

[135] Darwin, "Open Source Streaming Server," http://developer.apple.com/opensource/server/streaming/, 2006.

[136] Empirix, "PacketSphere Network Emulator," http://www.empirix.com/, 2005.

[137] M. Carson and D. Santay, "Nist net: a linux-based network emulation tool," *SIGCOMM Computer Communications Review*, vol. 33, no. 3, pp. 111–126, 2003.

[138] Endace, "DAG Network Monitoring Cards," http://www.endace.com/, 2005.

[139] Libpcap, "Packet Capture Library," www.tcpdump.org, 2006.

[140] TCPReplay, "PCAP Editing and Replay Tools for *NIX," http://tcpreplay.sourceforge.net, 2006.

[141] R. Love, "Introducing the 2.6 Kernel," *Linux Journal*, vol. 2003, no. 109, p. 2, 2003.

[142] VLC, "Videolan Streaming Client," http://www.videolan.org, 2006.

[143] LIVE555 Streaming Media, "C++ libraries for multimedia streaming," http://www.live555.com/liveMedia, 2006.

[144] ffmpeg, "libavcodec: open-source codec library," http://sourceforge.net/projects/ffmpeg/, 2006.

[145] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG) , "Reference Software Version 10.1 (JM 10.1)," http://iphome.hhi.de/suehring/tml/download/, 2005.

[146] MAPI, "Monitoring API," http://mapi.uninett.no, 2005.

[147] MPEG4IP, "Open source mpeg-4 streaming," http://mpeg4ip.sourceforge.net, 2005.

[148] W. v. Hagen, "Real-time and Performance Improvements for the 2.6 Linux Kernel," *Linux Journal*, vol. 2005, no. 134, p. 8, 2005.

[149] M. Lombardi, "Computer time synchronization," National Institute of Standards And Technology, Time and Frequency Division, Tech. Rep., October 2006, available at http://tf.nist.gov/timefreq/service/time-computer.html.

[150] S. Wenger, "Common Test Conditions for Wire-line Low Delay IP/UDP/RTP Packet Loss Resilient Testing," VCEG-N79r1.doc (Input document to ITU-T SG.16 VCEG), Sept. 2001. [Online]. Available: http://ftp3.itu.ch/av-arch/video-site/

[151] ——, "Error Patterns for Internet Experiments," Q15-I16r1, (Input document to ITU-T SG.16 VCEG), Oct. 2003. [Online]. Available: http://ftp3.itu.ch/av-arch/video-site/

[152] G. Roth, R. Sjöberg, G. Liebl, T. Stockhammar, V. Varsa, and M. Karczewicz, "Common Test Conditions for RTP/IP over 3GPP/3GPP2," VCEG-M77.doc, (Input document to ITU-T SG.16 VCEG), Apr. 2001. [Online]. Available: http://ftp3.itu.ch/av-arch/video-site/

[153] C. Calafate, M. Malumbres, and P. Manconi, "Performance of h.264 compressed video streams over 802.11b based manets," in *Proceedings of the 24th International Conference On Distributed Computing Systems Workshop*, 2004, pp. 776–781.

[154] D. Lin and R. Morris, "Dynamics of Random Early Detection," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*. Cannes, France: ACM Press, 1997, pp. 127–137.

[155] M. Yajnik, M. Sue, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, vol. 1, New York, NY, Mar. 1999, pp. 345–352.

[156] Digital Cinema Initiatives (DCI) and The American Society of Cinematographers (ASC), "StEM Mini-movie Access Procedyre," Available at http://www.dcimovies.com, Nov. 2004. [Online]. Available: http://www.dcimovies.com/

[157] Y.-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The Error Concealment Feature in the H.26L Test Model," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, Sept 2002, pp. 729–732.

[158] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of service support in IEEE 802.16 networks," *IEEE Network*, vol. 20, no. 2, pp. 50–55, 2006.

[159] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "IEEE standard 802.16: a Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 98–107, 2002.

[160] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen, "Broadband Wireless Access with WiMax/802.16: Current Performance Benchmarks and Future Potential," *IEEE Communications Magazine*, vol. 43, no. 2, pp. 129–136, 2005.

[161] C. Hoymann, "Analysis and Performance Evaluation of the OFDM-based Metropolitan Area Network IEEE 802.16," *Computer Networks*, vol. 49, no. 3, pp. 341–363, 2005.

[162] I. Koffman and V. Roman, "Broadband Wireless Access Solutions based on OFDM Access in IEEE 802.16," *IEEE Communications Magazine*, vol. 40, no. 4, pp. 96–103, 2002.

[163] WiMAX Forum^{TM} Regulatory Working Group, "Initial Profiles and the European Regulatory Framework," Sept. 2004. [Online]. Available: http://www.wimaxforum.org

[164] F. Wang, A. Ghosh, R. Love, K. Stewart, R. Ratasuk, R. Bachu, Z. Qing, and S. Yakun, "IEEE 802.16e system performance: Analysis and simulations," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 2, 2005, pp. 900–904.

[165] T. Schierl, T. Wiegand, and M. Kampmann, "3GPP compliant adaptive wireless video streaming using H.264/AVC," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2005, pp. 696–699.

[166] T. Schierl, H. Schwarz, D. Marpe, and T. Wiegand, "Wireless broadcasting using the scalable extension of H.264/AVC," in *Proceedings of the IEEE*

International Conference on Multimedia and Expo (ICME), 2005, pp. 884–887.

[167] D. T. Nguyen and J. Ostermann, "Streaming and congestion control using scalable video coding based on H.264/AVC," *Journal of Zhejiang University - Science A, and Proceedings of the International Packet Video Workshop*, vol. 7, no. 5, pp. 749–754, 2006.

[168] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on svc," *IEEE Transactions on Circuits and Systems for Video Technology, to appear*, p. Draft, Summer 2007.

[169] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end tcp-friendly streaming protocol and bit allocation for scalable video over wireless internet," vol. 22, no. 4, pp. 777–790, 2004.

[170] O. I. Hillestad, A. Perkis, V. Genc, S. Murphy, and J. Murphy, "Delivery of On-Demand Video Services in Rural Areas via IEEE 802.16 Broadband Wireless Access Networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Multimedia Networking and Performance Modeling*, H. Alnuweiri and R. Araujo, Eds., vol. 1. Torremolinos, Malaga, Spain: ACM Press, October 2–6 2006, pp. 43–51.

[171] ——, "Streaming of H.264/MPEG-4 SVC Video over IEEE 802.16 Wireless Broadband Networks," in *To be submitted.*, 2007.

[172] J. Mo, R. J. La, V. Anantharam, and J. Walrand, "Analysis and Comparison of TCP Reno and Vegas," vol. 3, 1999, pp. 1556–1563 vol.3.

[173] S. Wang and Y. Lin, "Nctuns network simulation and emulation for wireless resource management," *Wiley Wireless Communications and Mobile Computing*, vol. 5, no. 8, p. 899–916, 2005.

[174] *IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems.* IEEE Std. 802.16.2-2004, 2004.

[175] H. Schwarz, M. Wien, and J. Vieron, "Joint scalable video model JSVM-5 software," JVT-R203 (Output Document from Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), Jan. 2006. [Online]. Available: http://ftp3.itu.ch/av-arch/jvt-site/

[176] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos, "Video acceptability and frame rate," *IEEE Multimedia*, vol. 2, no. 3, pp. 32–40, 1995.

[177] A. Leontaris and P. C. Cosman, "Drift-resistant SNR Scalable Video Coding," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2191–2197, 2006.

[178] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Tech. Rep., September 1984.

[179] O. I. Hillestad, "Examples of resulting visual quality," Web page at Q2S, NTNU, Oct. 2006. [Online]. Available: http://www.q2s.ntnu.no/∼hillesta

[180] A. Acharya and J. Saltz, "A study of internet round-trip delay, Tech. Rep. CS-TR-3736, 1996. [Online]. Available: citeseer.ist.psu.edu/acharya96study.html

Appendix I

Test Material

I.1 StEM - Standardized Evaluation Material

The original StEM material is available in 4K resolution (4096 by 1714 pixels), has a temporal resolution of 24 frames per second and contains 16605 frames in total.

Figure I.1: Frames from the STEM sequence.

Fig. I.1 shows frame number 446 from the first scene of the STEM sequence. The scene has four scene changes, cross-fades, camera panning, complex object motion and a high level of texture detail, thus providing a good challenge for networked video applications in terms of coding efficiency, error resilience and concealment capabilities.

I.1.1 Characteristics of the StEM short movie

"StEM" consists of seven different parts. First, there is a synthetically generated introduction, followed by five scenes, and finally the closing credits.

1. Calibration sequence and introduction text
2. Introduction and wedding scene in daylight (STEM_A)
    - Frame 1113-1695: synthetic logo and introduction.
    - Frame 1695-1715: cross-fade into next scene.
    - Frame 1716-1895: scene 1, close-up of bride and guest.
    - Frame 1896-1914: cross-fade into next scene.
    - Frame 1915-1999: scene 2, juggler in the street.
    - Frame 2000-2018: cross-fade into next scene.
    - Frame 2019-2066: scene 3, woman on balcony.
    - Frame 2067-2086: cross-fade into next scene.
    - Frame 2087-2384: scene 4 with wedding procession.
    - Frame 2385-2543: scene 5, bicycle moving by, camera panning and zooming.
    - Frame 2544-3069: scene6, wedding procession, camera panning and zooming.
    - Frame 3070-3266: camera flash, black framing and fade out.
3. "Magic hour" evening scene (STEM_B)
    - Frame 3267-3383: title on black background fading into next scene.
    - Frame 3384-3466: people approaching dinner table.
    - Frame 3467-3660: man and woman waiting for wedding procession.
    - Frame 3661-3773: man on bicycle and wedding procession.
    - Frame 3774-3822: woman serving drinks.
    - Frame 3823-4015: wedding party arrives.
    - Frame 4016-4153: wedding party from above.
    - Frame 4154-4245: camera flash and black still frame.
    - Frame 4246-4287: cross-fade into next scene.
    - Frame 4288-4438: wedding party.
    - Frame 4439-4460: cross-fade

- Frame 4461-4676: wedding party.
- Frame 4677-4880: camera flash, black framing and fade out.

4. "Warm night" (STEM_C)

   - Frame 4881-5006: title on black background fading into next scene.
   - Frame 5007-5276: man and woman by fountain.
   - Frame 5277-5571: man and woman photographed (flash).
   - Frame 5572-5838: night shoot of wedding procession.
   - Frame 5839-5886: man and woman at table.
   - Frame 5887-6041: bride and groom arriving.
   - Frame 6042-6339: wedding party at night.
   - Frame 6340-6899: table with drinks, bride picking up wine.
   - Frame 6900-7015: wedding party.
   - Frame 7016-7366: toast, fading out.

5. "Cool night" (STEM_D)

   - Frame 7367-7543: title on black background fading into next scene.
   - Frame 7544-7756: couple by fountain, cross-fades, foggy night scene.
   - Frame 7757-8191: couple by fountain, wedding procession arriving.
   - Frame 8192-8280: cross-fade into next scene.
   - Frame 8281-8441: wedding procession arriving at party.
   - Frame 8442-8796: wedding party at night.
   - Frame 9797-8815: cross-fade
   - Frame 8816-9042: toast, close-up
   - Frame 9043-9061: cross-fade
   - Frame 9062-9135: toast from different angle
   - Frame 9136-9154: cross-fade
   - Frame 9155-9337: man walking, circular transition.

6. "And then comes the rain" (STEM_E)

   - Frame 9338-9487: title on black background fading into next scene.
   - Frame 9488-9734: rain on street with cross-fades
   - Frame 9735-10494: rainy street with people crossing.
   - Frame 10495-10597: cross-fade
   - Frame 10598-10770: rain on street fading out.

7. "Closing credits"