Bandwidth Efficient IPTV Distribution

– On Error Resilience and Fast Channel Change

Ulf Jennehag

Mittuniversitetet
MID SWEDEN UNIVERSITY

Department of Information Technology and Media
Mid Sweden University

Doctoral Thesis No. 39
Sundsvall, Sweden
2007

Mittuniversitetet
Informationsteknologi och medier
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-85317-77-6
ISSN 1652-893X

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie doktorsexamen torsdagen den 17 januari 2008 i L111, Mittuniversitetet, Holmgatan 10, Sundsvall.

Tryck: Tryckeriet Mittuniversitetet

To my wife

Abstract

Television is now changing its traditional distribution forms to being distributed digitally over broadband networks. The recent development of broadband Internet connectivity has made the transition to Internet Protocol Television (IPTV) possible. When changing distribution technique of an existing service, it is important that the new technique does not make the service worse from the user's point of view. Although a broadband network offers high capacity and has excellent performance there will be occasional packet losses and delays which could negatively influence the user experience of the delivered broadband service. Since bandwidth is a key constraint for video distribution there is a strong incentive for finding schemes to increase bandwidth utilization, especially when distributing high bandwidth IPTV services. In digital video coding it is common to use predictive coding to remove temporal redundancy in video sequences. This technique greatly increases the coding efficiency but makes the sequence more sensitive to information loss or delay. In addition, the use of predictive coding also introduce a inter frame dependency which could make the channel change significantly slower.

This thesis addresses two important areas related to bandwidth efficient IPTV distribution, namely error resilience and fast channel change. A method to numerically estimate the decoded objective video quality of scalable coded video is presented and evaluated. The method can be used to estimate objective video quality for a scalable video transmission system subject to packet-loss. The quality gain of temporally scalable video in a priority packet dropping environment is also investigated and quantified. Synchronization Frames for Channel Switching (SFCS) is proposed as a method to code and distribute video with IP-multicast, which can be used to efficiently combat packet-loss, increase bandwidth utilization, and offer a channel change speed up. The performance of SFCS is analyzed and bandwidth estimation expressions are formulated, analytical results are complemented with computer simulations. The results show that SFCS deployed in an IPTV delivery system can significantly lower the bandwidth consumption and speed up the channel change.

Sammanfattning

De traditionella distributionssätten för tv byts rask takt ut mot digital distribution via bredband. Det är den senaste utvecklingen av bredbandsnät som möjligtgjort lanseringen av nya bredbandiga tjänster, till exempel ip-tv. När distributionstekniker byt ut till förmån för nya är det viktigt att de nya teknikerna inte på verkar den underliggande tjänsten på ett negativt sätt. Även om nya bredbandsnäten oftast har mycket hög kapacitet och utmärkta prestanda så händer det att paket förloras eller blir fördröjda. Eftersom bandbredd är en flaskhals för distribution av video så finns det starka incitament för att finna effektivare distributionssätt, speciellt vid distribution av ip-tv med hög bandbredd. Vid kodning av digital video är det vanligt att använda prediktiv kodning för att reducera temporal redundans som är vanligt förekommande i videosekvenser. Användandet av prediktiv kodning ökar kodningseffektiviteten avsevärt men gör den kodade sekvensen känsligare för informationsförluster och fördröjningar. Prediktiv kodning av video skapar även ett beroende mellan olika bilder i sekvensen vilket kan göra att kanalbyten blir långsammare.

Avhandlingen omfattar två viktiga områden rörande distribution av ip-tv, nämligen motståndskraft mot paketförluster och snabba kanalbyten. En metod för att numeriskt förutsäga den objektiva videokvaliteten för skalbara videoströmmar har presenterats och utvärderats. Den föreslagna metoden kan användas för att ge ett värde på den förväntade objektiva videokvaliteten i ett skalbart videotransmissionssystem som är utsatt för paketförluster. Den kvantitativa kvalitetsökningen undersöks för temporalt skalbar video signal i ett system där viktigare skikt prioriteras i händelse av paketförluster. Vidare föreslås Synchronization Frames for Channel Switching (SFCS) som en metod för att distribuera digital video med IP-multicast. SFCS kan användas för att bätte hantera paketförluster, förbättra bandbreddsutnyttjandet samt minska kanalbytestider. Prestanda för metoden analyseras och genom analytiska beräkningar för bandbreddsutnyttjande kompletterade med datorsimuleringar. Resultaten visar att användningen av SFCS i ett distributionssystem för ip-tv kan minska bandbreddsutnyttjandet signifikant samt reducera tiden för kanalbyten.

Acknowledgements

Firstly I would like to thank my supervisor Docent Tingting Zhang and my secondary supervisor Dr. Stefan Pettersson, without whose support, guidance, and commitment, this work would have been impossible. I would also like to thank Professor Björn Pehrson and Professor Youzhi Xu who made it possible for me to start my Ph.D studies.

I would like to also thank Dr. Patrik Österberg, Lic.Eng Daniel Forsgren and M.Sc Hans Eric Sandström for being the best working colleagues and friends one can wish for.

I also like to thank Dr. Mikael Gidlund for giving me suggestions for improvement of this thesis and also for making it possible for me to spend almost half a year being employed at Acreo AB in Hudiksvall.

I would also like to extend gratitude toward all the people outside of the university supporting our research in the MUCOM group, especially Karin Nygård-Skalman who has been helping us with applications for funding and extending our network of industrial partners.

I also like to thank all my colleagues in MUCOM and IKS department for your company and interesting discussions in the coffee room.

A big thank to my mother, father and brothers. This work is based on your endless and unconditional support and love.

At last and most, to my wife Johanna. Without your love and support this work would have been impossible. I dedicate this thesis to you.

Contents

List of Papers

This thesis is based mainly on the following papers, herein referred by their Roman numerals:

I Tingting Zhang, Ulf Jennehag, and Youshi Xu , Numerical modeling of transmission errors and video quality of MPEG-2. In *Signal Processing: Image Communication*, Volume 16, Issue 8, Pages 817-825, May 2001.

II Daniel Forsgren, Ulf Jennehag, and Patrik Österberg , Objective End-to-End QoS Gain from Packet Prioritization and Layering in MPEG-2 Streaming. In *Proceedings of the 12th International Packet Video Workshop (PV 2002)*, Pittsburgh PA, USA, April 2002.

III Ulf Jennehag and Tingting Zhang , Increasing Bandwidth Utilization in Next Generation IPTV Networks. In *Proceedings of the 2004 International Conference on Image Processing (ICIP 2004)*, Volume 3, Pages 2075 - 2078, Singapore, October 2004.

IV Ulf Jennehag, Tingting Zhang, and Stefan Pettersson , Improving Transmission Efficiency in H.264 Based IPTV Systems. In *IEEE Transactions on Broadcasting*, Volume 53, Issue 1, Pages 69-78, March 2007.

V Ulf Jennehag and Stefan Pettersson , On Synchronization Frames for Channel Switching in a GOP-based IPTV Environment. In *Proceedings of the fifth IEEE Consumer Communications & Networking Conference (CCNC 2008)*, Las Vegas NV, USA, January 2008.

List of Figures

List of Tables

Terminology

Abbreviations and Acronyms

AAC	Advanced Audio Coding
CABAC	Context-adaptive binary arithmetic coding
ADC	Analogue-Digital Converter
DCT	Discrete Cosine Transform
DS	Differentiated Services
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DTS	Decoding Time Stamp
DVB-T	Digital Video Broadcasting, Terrestrial
FCC	Fast Channel Change
HDTV	High Definition Television
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPSTB	Internet Protocol Set-Top-Box
IPTV	Internet Protocol Television
ISMA	Internet Streaming Media Alliance
ISO	International Organization for Standardization
live-TV	Traditional channel based television service, with a fixed timeline.
MPEG	Moving Picture Experts Group
OSI	Open Systems Interconnection
PTS	Presentation Time Stamp
RAP	Random Access Point
RTP	Real-time Transport Protocol
SDTV	Standard Definition Television
TCP	Transport Control Protocol
TS	MPEG-2 Transport Stream
UDP	User Datagram Protocol
VoD	Video on Demand

Mathematical Notations

fps	frames per second
M	number of available channels
N	total number of clients
d	average duration between resynchronization requests
N_m	number of clients tuned to channel m
P	size of a Inter-frame i bits
R_{RSM}	RAP_{SYNC} rate
R_{PSM}	inter-frame rate SFCS main
R_{RG}	RAP_{GOP} rate
R_{PG}	inter-frame rate GOP
r_{RSS}	size ratio RAP_{SYNC}/P
r_{RG}	size ratio RAP_{GOP}/P
B_{GRCL}	expected GOP bandwidth in one RCL
$B_{GRRL,m}$	expected GOP RRL bandwidth for one channel m
B_{GRRL}	expected GOP RRL bandwidth for all channels
B_{SRCL}	expected SFCS bandwidth in one RCL
$B_{SRRL,m}$	expected SFCS RRL bandwidth for one channel m
B_{SRRL}	expected SFCS RRL bandwidth for all channels

Chapter 1

Introduction

Over the last fifteen years there have been continuous developments associated with the Internet. More people have gained Internet access as well as making use of the services offered. The increasing number of new users and services offered has caused a surge for connections with high bandwidth, low delay, and high reliability. This type of network connections called broadband connections, have increased greatly in numbers over the years. The increased capacity in the network allow for new and exciting services. Streaming media is one such group of services that has grown rapidly growing over the last few years. Internet Protocol Television (IPTV) is a specific example. IPTV specifies a group of multimedia services including, live-TV, Video on Demand, and Pay-per view.

Multimedia services such as IPTV rely heavily on streaming video techniques. For a streaming video service to be at all feasible, it must utilize compression techniques in order to reduce the amount of data being transmitted. Modern compression techniques utilize predictive coding which makes the stream sensitive to information loss. Since streaming video is a real-time service it is also sensitive to information being delayed or received out of order.

Some of the current deployments of IPTV suffer from slow channel change performance. This is in some part due to the long waiting time for the decoder to find a suitable starting point in the new stream.

In summation, IPTV distribution requires a high performing network infrastructure, information loss probabilities and network delay variance needs to be very low otherwise the service will not function properly.

This thesis deals with two of the issues regarding IPTV distribution, namely, error resilience and fast channel change.

1.1 Background and Problem Motivation

The development of network infrastructure is still increasing and more and more households are gaining broadband access. There is no actual figure defining the capacity of a broadband connection, but the common notion of a broadband connection is that it is "always on" and often offered to a flat rate. A broadband connection is considered to include Digital Subscriber Line (DSL) connections but excludes modem and single ISDN connections. Sweden is a good example of a market where there has been a rapid expansion in broadband development. In their annual report "Bredband i Sverige" the "Post och Telestyrelsen (PTS)" in Sweden have investigated the number of Swedish households with the means to acquire a broadband connection [1] [2] [3]. Over the years the development has been steady and rising. In their last report PTS indicated that over 97% of all Swedish households have the means of acquiring a broadband connection.

This broadband infrastructure development surely increases the possibility of successfully deploying services with high bandwidth demands. Internet Protocol Television (IPTV) [4] [5] [6] [7] is one such service. IPTV is now gaining popularity very rapidly, Informa Research [8] state that the market will grow by a factor of seven by 2011 based on the 2006 numbers. The notion of IPTV being a rapid increasing industry is also true in the broadband dense country of Sweden [9]. For example, two major Swedish ISPs, TeliaSonera [10] and Bredbandsbolaget [11], are now promoting IPTV services on their homepages and in broadcast TV commercials.

Distributing multimedia demands high bandwidth. Increased bandwidth means higher quality or more content distributed over the network. The demands for low bandwidth and high quality drives the media compression industry which are relentlessly researching new and more complex algorithms. Bandwidth issues is always going to be a interesting and challenging topic in IPTV research.

Distributing multimedia means that there are high demands placed on the network. Multimedia is sensitive to information loss and excessive delay jitter due to the predictive coding used in modern multimedia compression techniques. For example, information loss in video data manifests itself by means of blocking artifacts in the image sequence. IPTV is very sensitive to packet-loss and require almost a error-free environment [12] to give excellent service quality.

Modern packet switched broadband networks often have excellent performance in terms of extremely low packet loss ratio and a low delay with low variance [13], this is, however, not the case for most networks, where occasional packets still are lost or delayed.

One important contributory factor to information loss in packet switched networks is congestion. As compressed multimedia often rely on predictive coding it is reasonable to assume that some information packets in the stream can be considered as being more important than other packets. In a packet loss situation the result of the lost packet can have different impacts depending on how important the packet lost was. In the situation associated with congestion, there can be beneficial for the congested routing node to throw away packets with low importance first. If a packet is lost it is also important to know how the receiver is able to recover from that condition without causing to much overhead in the network. Bearing this in mind, it is possible to argue that it is important to have methods which are able to combat this low frequency packet loss, which is often caused by congestion.

Another issue in IPTV distribution is the increased channel change time compared to analog television distribution. This is caused by several factors and in which the time to wait for a suitable entry for the decoder in the stream must be considered as being one of the biggest. A reduction in IPTV channel switching time targets an important area of IPTV research.

1.2 Overall Aim

This thesis will address IPTV distribution over networks. More specifically it will address error resilience issues for video transmission in terms of objective video quality and bandwidth consumption. It also targets the performance of channel switching in terms of bandwidth efficiency, tune-in time, and objective video quality.

1.3 Scope

By definition multimedia content contains more than one media type. Audio and video are two types of media that are often combined to represent a course of events. Modern audio compression techniques achieve such good results [14] that the audio signal only occupy a fraction of the bandwidth compared to the video signal. Therefore, the studies presented in this thesis are focused on different aspects of broadcast quality video distribution over IP networks. The underlying physical media used for packet transport in the network is out of scope for this thesis.

1.4 Concrete and Verifiable Goals

The thesis deals with two different tracks related to IPTV distribution.

1.4.1 Error Resilience in Video Transmission

Within the area of error resilience in video transmission, three different subareas have been studied.

- Due to predictive coding of video sequences it is reasonable to assume that some information is more important than other. Packets containing video data can be classified into layers by the importance of the data residing in them. In this case, the lower layers predict the higher layers. By calculating the effect that a packet loss in one layer has on the others, suitable schemes for error resilience and protection can be constructed.

- Since the main reason for packet loss is congestion it makes sense to allow routers to utilize information concerning the importance of the data in one packet in relation to other packets in the stream video stream. By assigning packets to different classes the router can, in a state of congestion choose to throw away packets with lower importance first in order to minimize the effect on the quality of the decoded video.
- When a video packet is lost it will probably influence the decoded result of other packets due to the use of predictive coding. It is important to stop this error propagation as early as possible. Therefore it is important to investigate video distribution solutions that address is area.

1.4.2 IPTV Fast Channel Change

The use of predictive coding and the spacing of key frames (aka random access points), influences channel switching time. It is of great interest to reduce the channel switching time without compromising steady-state compression performance. This thesis addresses a tune-in channel approach based on one steady state stream accompanied by a tune-in stream distributed on two separate multicast addresses. A channel tune-in is performed by first joining and receive tune-in information form the tune-in stream and later join the steady state stream. The information from the tune-in stream is spliced with the steady state information and is then fed to the decoder.

- One of the incentives to use fast channel change techniques is to reduce bandwidth consumption. Therefore it is of great interest to investigate methods to estimate the bandwidth consumption caused by fast channel change techniques.
- The primary purpose of a fast channel change technique is to reduce the tune-in time when changing channels. Tune-in time estimations and the implications associated with the bandwidth are of great interest.
- When splicing two streams from different encodings there is a risk of experiencing quality distortions caused by the differences in the reference pictures. Methods to reduce this distortion or remove it completely are of great interest.

1.5 Outline

In chapter 2 the reader is introduced to the area of multimedia coding and transmission as well as related aspects. Chapter 3 introduces the reader to the concept of the developed side-steam tune-in technique which is a major part of the thesis. Chapter

4 presents related work in the field of error resilience in video transmission and the contributions by the author in the area. Chapter 5 introduces the reader to the field of fast channel change in IPTV distribution, related work, and the contributions by the author in the field. A summary is given in chapter 6 together with suggestions for future work in the two fields presented.

1.6 Contributions

The author's contributions to the area are provided in the included published papers. The main contributions are stated here.

A study in which the mathematical relationship between packet loss and the decoded video quality is presented and evaluated, in the case of temporally layered MPEG-2 transmitted over a simple network experiencing packet-loss. The analytical results are verified using software simulations. The results are presented in paper I. The author has been sorely responsible for the design and implementation of the simulation environment, including development of the video quality measurement method.

Paper II includes a study of the objective video quality gain attained by using temporally layered MPEG-2 streaming, when applying priority dropping techniques at intermediate router nodes. Different queuing strategies and the influence of packet size are also investigated. The author contributed by participating in the design and construction of the simulation environment and the video quality measure.

Paper III-V deal with a idea regarding the use of side stream information for faster channel tune-in and error recovery in an IPTV environment. A complete mathematical analysis of the system is presented and the results are confirmed by means of a simulation. The method is applied to both traditional encoder variants and more complex ones using switching frames. The presented method can be used to reduce the channel switching time and to combat error propagation caused by packet loss. In paper V a measure of tune-in quality for the system applied to a group-of-pictures environment is presented. In these publications the author has been responsible for all the results with some additional assistance from Doc. Tingting Zhang with regards to the development of the mathematical bandwidth estimation expressions. Dr. Stefan Pettersson provided assistance in the form of suggestions which were mainly concerned with the writing of the papers and how to present the information in an understandable manner.

Chapter 2

Background Theory

This chapter introduces the reader to the field of image and video representation, video compression, audio compression, multimedia packetization, and the transmission of multimedia over IP networks.

2.1 Digital Images

Digital images can be visualized as a collection of color data (values) arranged in a matrix of small pixels. The actual value for one given pixel position within the matrix represents the color of that pixel. If the pixels are sufficiently small, the human eye will perceive the image as smooth and consistent. Several images in a sequence can be considered as being a video. Images in a video sequence are often captured at a constant frequency where each of the images represents a snapshot of the scene captured. The images in a video sequence are also called frames or pictures.

2.1.1 Color Representation

There are several standards for representing colors in digital images and digital video. For example; RGB: Uses a fixed number of bits for the each one of the three color channels R (red), G (green) and B (blue). This format is often used in computer applications. YC_BC_R: Is component based color representing scheme. Y represents luminance value of the pixel and the C_B C_R represents the chrominance value, one for each channel. This component system has several modes each with different sampling ratios with regards to the chrominance information. The 4:4:4, 4:2:2 ,and 4:2:0 sampling ratios and positions are outlined in Figure 2.1. 4:4:4 Sampling ratio: In this case, the Y and the C_B C_R information and are present for every pixel. 4:2:2 Sampling ratio: In this case, Y is present in every pixel and C_B C_R information is present for every second pixel in the horizontal direction. This gives a color compression of 1/3 compared with the 4:4:4. 4:2:0 Sampling ratio: The 4:2:0 format further reduces

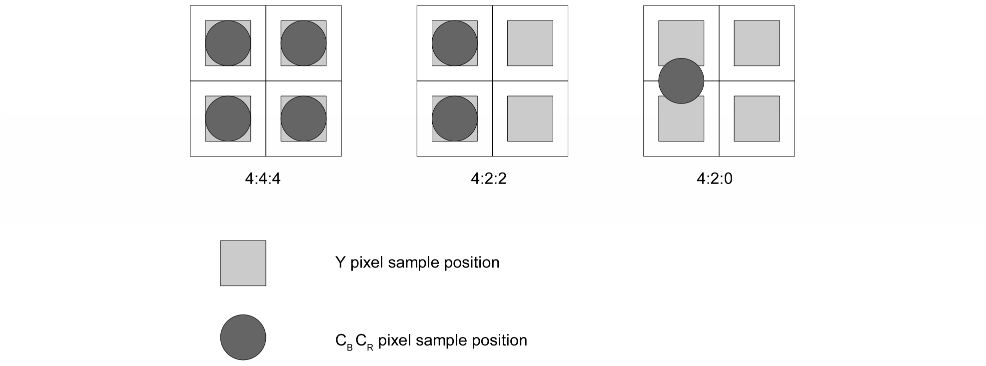


Figure 2.1: Component sampling in YC_BC_R formats

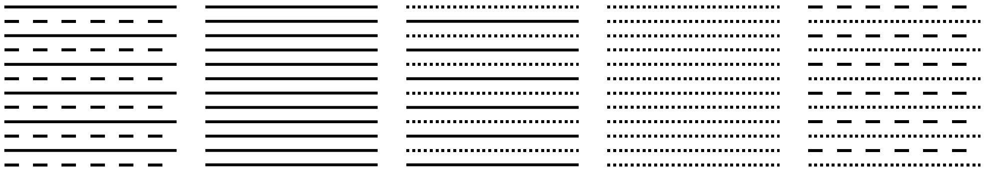


Figure 2.2: Interlace using alternating odd/even field update

the number of chrominance samples. Chrominance information is only available every fourth pixel, where the chrominance values are calculated as the mean of two pixels, as illustrated in figure 2.1.

2.1.2 Interlace

Interlace scanning [15][16] is a widely used method to increase the perceived video picture update frequency and reduce the information usage. In general interlace means that temporal resolution can be increased through a reduction in the vertical resolution. The most common variant of interlace is based on updating either the odd or the even lines, called fields, on each update of the viewing screen, as illustrated in Figure 2.2. Two fields make up a picture. To further enhance the viewing experience the two fields in a picture are sampled at different times. In contrast to interlace scanning, progressive scanning is often mentioned. In progressive scanning all lines are updated every time. Interlaced pictures are used both in analog and digital video transmission.

2.1.3 Resolution Formats

A large number of screen resolution variations associated with television and multimedia streaming formats exist. These include QCIF, CIF, 576i, 720p, 1080i ,and 1080p. Where *i* denotes interlace scanning and *p* denotes progressive scanning. 576i is the

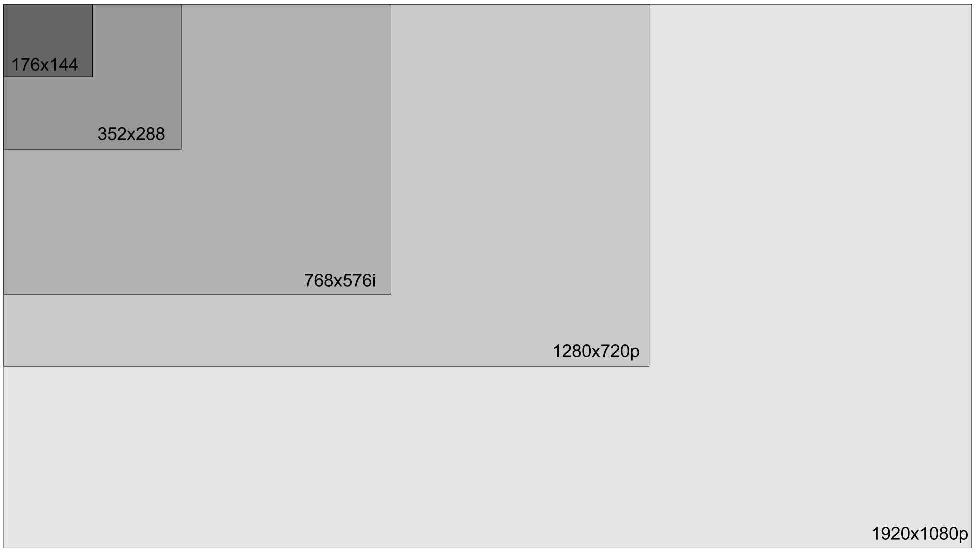


Figure 2.3: Different common screen resolutions compared

same as PAL resolution and is also referred to as SDTV. The format 720p is called HDTV, or half-HD, in contrast, the 1080i and 1080p are called Full-HD. The Full-HD fomat is specified by EBU in [17]. The size ratio between the different resolutions are presented in figure 2.3.

2.2 Video Coding

All common implementations of video encoder/decoders (codecs) rely on several data structures as illustrated in figure 2.4 to aid in the coding process of the video. A picture is typically divided into blocks, which are generally of the size of 8 by 8 pixels. The block is then subject to transform coding. Blocks are often grouped together in macroblocks. A common macroblock constellation is that four horizontal and vertical adjacent blocks form a macroblock. Motion estimation/compensation is often performed at the macroblock level. A group of macroblocks makes a slice. The slice structure often contains the means for the decoder to resynchronize with the data stream, which is achieved by inserting a unique bit sequence called a start code. A coded picture consists of one or several slices. A picture can be encoded either *intra* (I) picture, utilizing information within the picture itself, or as an *inter* (P/B) -picture, using information from adjacent pictures in time (temporal prediction). Section 2.2.1 and 2.2.2 describes inter and intra coding in more detail. Pictures are typically arranged into a Group Of Pictures (GOP) where the first picture in the GOP is *intra* coded.

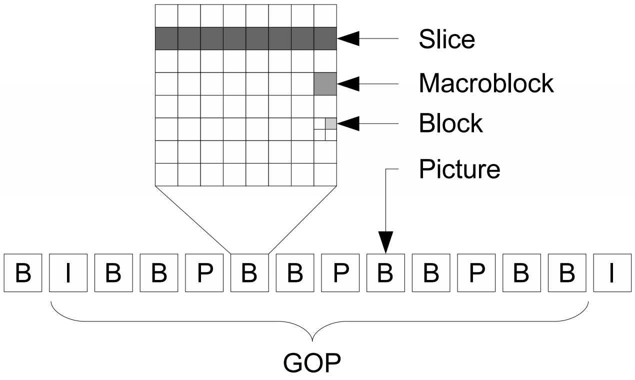
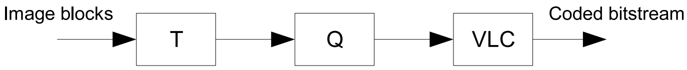
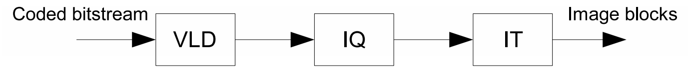


Figure 2.4: Common data structures used in video compression



(a) Intra encoder



(b) Intra decoder

Figure 2.5: Intra encoding and decoding

2.2.1 Intra Coding

Coding of pictures with no reference to other temporal information is called *intra* coding. *Intra* coded pictures are often referred to as I-pictures or I-frames. *Intra* coders utilize the spatial redundancy that exists in the picture to be coded. Figure 2.5 illustrates a simple intra coding process. Image blocks, e.g. 8X8 matrices of pixels, are transformed using a frequency transform, e.g. Discrete Cosine Transform (DCT) function. The transform coefficients are quantized, zig-zag scanned (Q), and Variable Length enCoded (VLC). The decoding is a reverse process of the encoding with a Variable Length Decoder (VLD) followed by an inverse quantize function (IQ) and an inverse DCT (IDCT). The Joint Photographic Experts Group (JPEG)[18] has developed a popular and widely spread *intra* coding standard utilizing a block-based variant of the intra encoder/decoder called JPEG.

2.2.2 Inter Coding

Since video can be interpreted as a sequence of pictures captured at a predetermined rate, there is a great deal of information that is more or less static between pictures. This redundancy can be removed to further increase the coding efficiency. figure 2.6

(a) Foreman picture 66 (b) Foreman picture 69 (c) Difference picture

Figure 2.6: Displaying previous picture, picture to be coded, and the difference picture

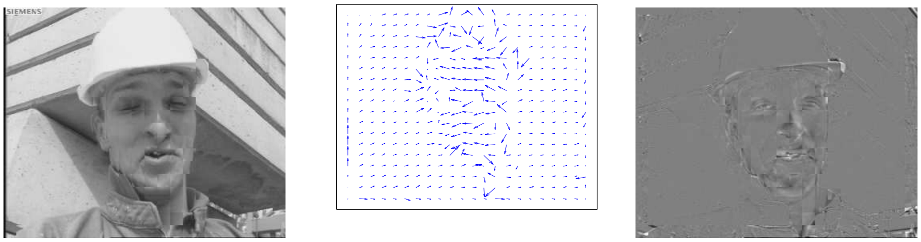
illustrates this by showing the pixel difference of two subsequent pictures (a) and (b). The uniform gray areas in 2.6 (c) correspond to areas with low temporal motion. The coding of picture sequences by removing temporal redundancy is called *inter* coding.

Inter coding relies on techniques to find matching information in the reference picture and reuses this information in the picture to be coded. This minimizes the amount of information required to describe the new picture thus increasing the compression ratio. Although this techniques may be related to such a new phenomenon as coding of digital images, the idea of reusing known information in moving pictures is actually quite old. As early as 1929 Ray Davis Kell introduced this idea in a patent [19].

This idea of reusing known information is implemented in the majority of modern video compression schemes. The technique of finding a suitable part of the reference picture to reuse is referred to as Motion Estimation (ME) [16]. The result of the ME process provides information relating to the reference picture the best match can be found. This directional information is called a *motion vector*. The reverse process, using the motion vector to copy information from the reference picture to the picture to be reconstructed is called Motion Compensation (MC). ME/MC is often performed at the macroblock level. In figure 2.7 (a) the motion compensated representation of the picture figure 2.6 (b) is displayed. This picture is reconstructed only with the previous picture figure 2.6 (a) and the motion vectors in Figure 2.7 (b).

This approach only requires that the motion vectors should be transmitted, which reduces the information for the reconstruction of the picture even further. However, there is the possibility that the result produced may not be perfect and this is illustrated in figure 2.7 (c) which contains the difference between the figure 2.6 (b) and 2.7 (a). This residue information must also be coded and transmitted in order to reconstruct a good representation of figure 2.6 (b). The residue is coded using similar techniques to those described in section 2.2.1

Inter coded frames or pictures are often referred to as P-frames or P-pictures. Pictures acting as a reference for other pictures, i.e. I and P-pictures, are collectively referred to as reference pictures. If the *inter* coded picture uses information from both the preceding and following reference pictures, it is referred to as a bi-directional predictive frame, or more concisely as a B-frame or B-picture. B-pictures are generally



(a) Predicted picture (b) Motion vectors (c) Difference picture

Figure 2.7: Displaying the predicted image, the motion vectors, and the residue image

not used to predict other pictures.

One obvious drawback associated with inter coding is the necessity to have a reference picture. Without the reference it is impossible to reconstruct a good representation of the coded picture. Bearing this in mind, reference pictures can be considered as being more valuable than pictures not used as a reference for other pictures. This is a fact that must be considered when transmitting *inter* coded frames over those mediums which are prone to information loss.

2.2.3 Hybrid Coding

When utilizing *intra* and *inter* coding in conjunction, the result is called a hybrid coder and most modern coders are variants of this method. Figure 2.8 illustrates a block scheme of a hybrid encoder and the details are explained in the following example. A combination of both *intra* and *inter* coding is involved in this encoding process. The first decision of the encoder (C) is the mode in which to code the current picture (side information). If the picture is to be coded as an *intra* picture the encoder takes the input block and then applies a transform function (T). The transformed coefficients are then quantized and zigzag scanned (Q). The result is then variably length coded (VLC) and this is then placed in the output buffer. If the picture is to be *inter* coded, a best match search by the motion estimator (ME) is performed on the reference picture found in the picture memory (PM) for the block to be encoded. The location of this block is called the motion vector. The residue between the input block and the match found is transformed (T), quantized, zigzag scanned (Q), and variably length encoded (VLC). The encoded coefficients are combined with the motion vector indicating the best match in the output buffer.

The decoding process of an intra picture can be described as a reverse process of the encoding, the hybrid decoder block scheme is shown in figure 2.9. Decoding an *intra* block is performed by means of a variable length decode of the coefficients, inverse quantize, and it is then run through an inverse transform function. The resulting image block is copied to the output image. Decoding an intra block is performed by copying (MC) the area from the picture memory (PM) indicated by the motion vector, decoding the residue, and finally adding the two together.

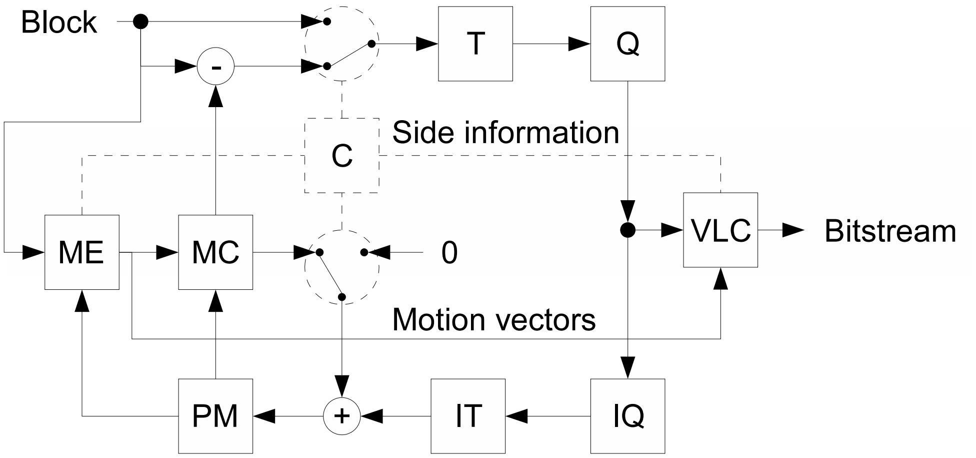


Figure 2.8: Hybrid encoder

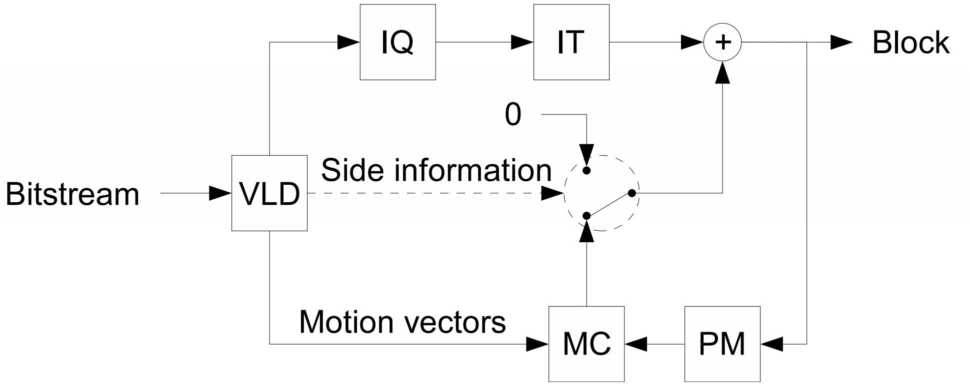


Figure 2.9: Hybrid decoder

2.2.4 SI/SP- frames

The difference between SP- and P-frames is that SP-frames allow identical frames to be reconstructed even when they are predicted using different reference frames.

SI/SP frames were proposed and introduced by Karczewicz and Kurceren [20] based on the work introduced by Färber and Girod [21]. SI/SP-frames are sometimes referred to as switching frames [22]. However, this is not mentioned in the paper by Karczewicz and Kurceren where the new frame types are simply called SI/SP-frames without any further explanation.

SP-frames are a variant of P-frames but have the property of being able to reconstruct identical frames with different reference frames. The following example further explains the properties of SP-frames and is illustrated in Figure 2.10. Stream A, which only contains *inter* coded material (P-, or B-pictures) is currently being

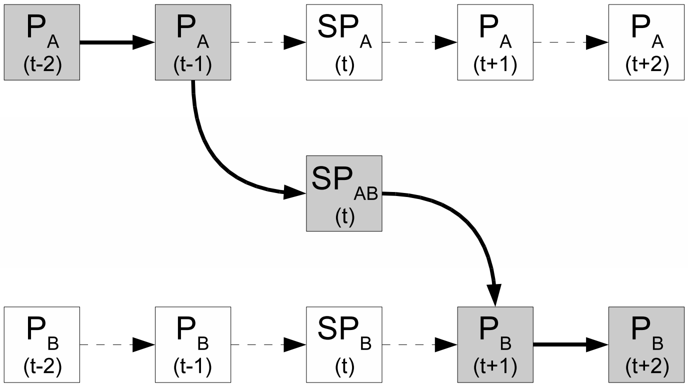


Figure 2.10: Switching between streams with SP-frames.

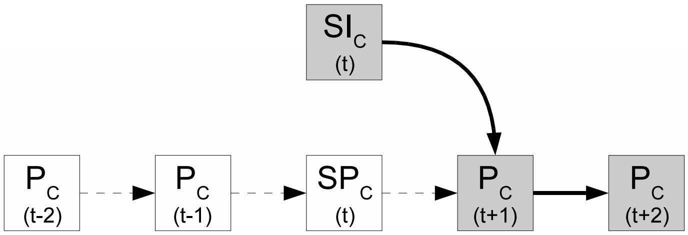


Figure 2.11: SI-frame acting as a stream entry point.

decoded, i.e. PA. A switch to the content in stream B is then performed, and the switching picture SPAB(t) is then decoded, instead of the SPB(t) picture. The decoded output of the SPAB(t) picture is exactly the same as the decoded output of the SPB(t) picture. This enables the next picture, i.e. PB(t+1) to use any of these two pictures as a basis for the prediction. Hence, a switch from steam A to stream B has been made utilizing the temporal redundancy provided by stream A, when decoding stream B. The switching picture must be coded using knowledge concerning both stream A and B.

SI-frames behave in exactly the same manner as a SP-frame but do not require a reference frame to be decoded. SI-frames can be used as stream entry points as illustrated in figure 2.11.

SI/SP-frames form part of the extended profile of the H.264 standard but is referred to, in that case, as SI/SP-slices since H.264 does not have explicit frame/picture structure, for more information see section 2.4.4.

2.3 Audio Coding

Single channel audio can be looked upon as a one dimensional time variant signal. Compression of audio is rather straightforward at a first glance but the imperfections of the human auditory system opens it up to many additional coding techniques [23]. For example, *temporal masking* is the description of the property of a strong sound preceding a weak sound which masks the weaker sound to such an extent that it is often inaudible. Another example is *frequency masking*, this means that a loud sound at a particular frequency cause weaker sounds to be masked which have a frequency close to that of the loud sound. This masking causes the weak sound to become inaudible to the human ear. The above mentioned properties and several others are utilized in most modern audio compression algorithms.

2.4 MPEG

Motion Pictures Expert Group (MPEG) is the commonly known name of the ISO/IEC working group ISO/IEC JTC1/SC29 WG11. MPEG consists of approximately 350 different people from both industry and educational bodies and was founded in 1988. The MPEG proposes and develops standards for multimedia coding and transmission.

2.4.1 MPEG-1

In 1993 the MPEG-1 [24] was introduced as an ISO/IEC standard. MPEG-1 specifies audio and video compression and the purpose is to enable the delivery of acceptable quality for the combined bitrate of 1.5Mbps. The complete MPEG-1 standard consists of five parts. Part 1, Systems, covers the issues associated with combining one or more streams into one stream suitable for retrieval and play out. The systems layer also provides the necessary timing information required to decode the combined stream. Part 2, Video, specifies the video compression layer. MPEG-1 Part 2, includes ME/MC techniques as well as support for B-frames. MPEG-1 does not support interlaced video. Part 3, Audio, specifies a compression scheme for mono and stereo audio signals at various bitrates. MPEG-1 part 3 includes three encoding decoding layers and in which layer 1 has a low complexity and compression ratio. Layer 3 is also known as *mp3* and uses a more complex coding model and thus achieves a better compression ratio at a specified bitrate compared to the two other specified layers. Part 4 specifies compliance testing of the implementations of the software and hardware for part 1-3 encoders/decoders. Part 5 is a software implementation of part 1-3. Gall summarizes the main features of MPEG-1 in [25].

2.4.2 MPEG-2

In 1996 the MPEG finalized and standardized a new suite for multimedia coding, MPEG-2. MPEG-2 gained international status when the work was accepted by the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) [26]. MPEG-2 targets standard definition television (SDTV) resolution but also includes high definition television (HDTV) resolution. Coding of interlaced signals was included in MPEG-2 which greatly improves perceived quality when it is displayed on a standard television set. The MPEG-2 standard also included updated audio encoding schemes and protocols for packetizing and distribution. Haskell *et.al* has published a book [15] covering the features of the MPEG-2 standard in a comprehensive manner.

MPEG-2 Part 1, Systems: Systems is a comprehensive framework for multiplexing and packetizing multimedia data suited to a wide range of communication channels and protocols. The MPEG-2 Systems standards text is shared with the International Telecommunication Union (ITU) standard H.222.0 [27].

Part 2, Video: Specifies compression of video signals targeted for SDTV and HDTV resolutions. MPEG-2 Video has support for compression of interlaced video such as PAL and NTSC signals. As with systems MPEG-2 Video shares standard text with the ITU standard H.262 [28].

Part 3, Audio: The audio part of MPEG-2 which has support for multi-channel audio coding but is also backward compatible with MPEG-1. Part 4, Conformance: As with MPEG-1 this part specifies compliance testing of the implementations of the software and hardware for the MPEG-2 part 1-3 encoders/decoders. Part 5, Software 5: Software implementations of part 1-3. Part 6, Digital Storage Media - Command and Control (DSM-CC): Specifies a set of protocols with functions to manage MPEG-1 and MPEG-2 streams. Part 7, NBC Audio: Multi-channel audio, not backward compatible with MPEG-1 Part 8, 10 bit Video: Support for 10bit video, this is no longer continued due to a lack of interest on the part of industry. Part 9, Real Time Interface: Specification of a real-time interface to transport a stream for adaptation to networks carrying transport streams. Part 10, DSM-CC Conformance 10: Conformance testing specification for part 6.

2.4.3 MPEG-4

MPEG-4 [29] was standardized at the beginning of 1999, and is a large framework for multimedia compression and communication. The video coding of MPEG-4 is specified in part two of the standard, which also divides into different profiles, and which includes the simple profile (SP) and the advanced simple profile (ASP). Several video compression techniques are described within the framework, e.g. block-based hybrid coding, B-pictures, object-based video coding, advanced audio coding to mention but a few. MPEG-4 ASP is also used under the name DivX [30] and has for some years been the dominant method for compressing video in order to fit a CD-ROM. MPEG-4 has a similar structure to that of both MPEG-1 and MPEG-2 but

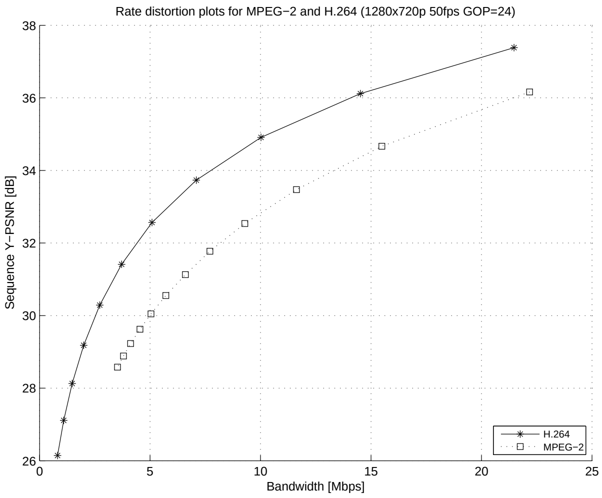


Figure 2.12: H.264 and MPEG-2 encodings of the SVT Fairytale sequence, 1280x720p 50fps

consists of approximately twenty different parts covered by the scope of the standard.

2.4.4 MPEG-4 AVC

In 2003 a new part of the MPEG-4 standards suite was completed which was called MPEG-4 Part 10, Advanced Video Coding (ISO/IEC 14496-10). MPEG-4 AVC is also known as H.264 [31]and was previously known as H.26L. MPEG4 AVC/H.264 was developed by the Joint Video Team (JVT) consisting of the ITU-T related workgroup Video Coding Experts Group (VQEG) and MPEG. The text in the MPEG-4 AVC standard is identical to that of H.264. MPEG-4 AVC/H.264 introduces a new set of video coding features, which increase the compression efficiency and flexibility compared to the previous versions of MPEG-4 and MPEG-2. H.264 offers a far superior compression ratio as compared to MPEG-2. In figure 2.12 a comparison of MPEG-2 and H.264 encodings of the SVT high definition progressive sequence "Fairytale" in resolution 1280x720 at 50 frames per second is presented. The open source software x264 [32] was used to encode the H.264 sequences and FFMPEG [33] was used to encode the MPEG-2 sequences. Both plots of the encodings have variable bitrate settings for constant quality during the sequence. The GOP size was set to 24 and two B-frames are used between P-frames. The reported objective quality is the PSNR of the whole sequence. It is worth noting that for a given quality of PSNR equal to 35.0 dB, H.264 has approximately 60% of the bandwidth as that for MPEG-2. For further reading with reference to H.264 the document by Wiegand et. al [34] is excellent and covers most of the aspects of H.264. Richardson has also written a book [22] covering the subject which is also well worth reading.

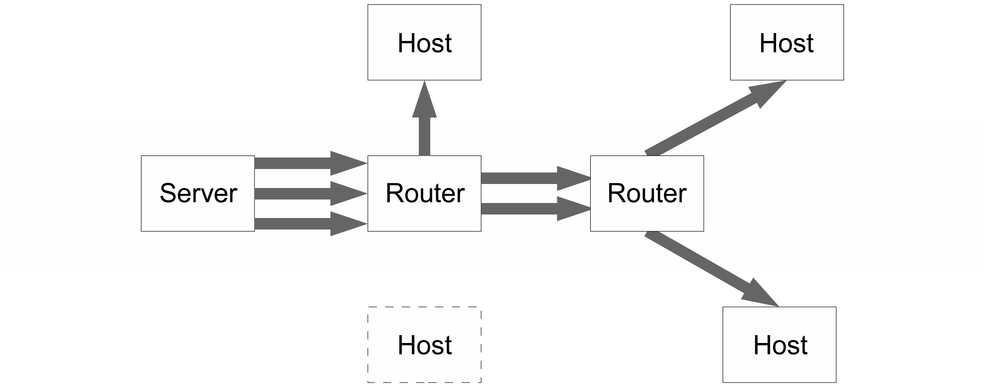


Figure 2.13: Unicasting packets from one host to other hosts.

2.5 Packet Switched Networks

Information distribution in a packet switched network is performed by enclosing the information to be distributed in a packet and marking the packet with the source and destination address. The source and destination of the packets are identified by unique addresses. Packets are distributed through the network by a set of intermediate nodes called routers that route the packets to their destinations.

This section will introduce the reader to the different transmission modes that exist in a packet switched network.

2.5.1 Unicast

Unicast transmission deals with one to one communication and includes for example, one host sending exclusive information to another host. However, it is also possible to use unicast for the occasions when one host wants to communicate the same information to several hosts. Figure 2.13 illustrates this with a server sending the same information to several hosts interested in this particular information. As can be seen in the figure, there is a substantial wastage of bandwidth in the shared links since the information is sent independently from the server to each one of the hosts.

2.5.2 Broadcast

Broadcast transmission mode is very useful when there is a requirement to send information to a large number of connected hosts. However, broadcasting information means that the information reaches all connected hosts, including those who have not requested the information. It is possible to distribute multimedia with a broadcast transmission mode but, as described above, it introduces an additional network load. Figure 2.14 illustrates a multimedia delivery situation using broadcast.

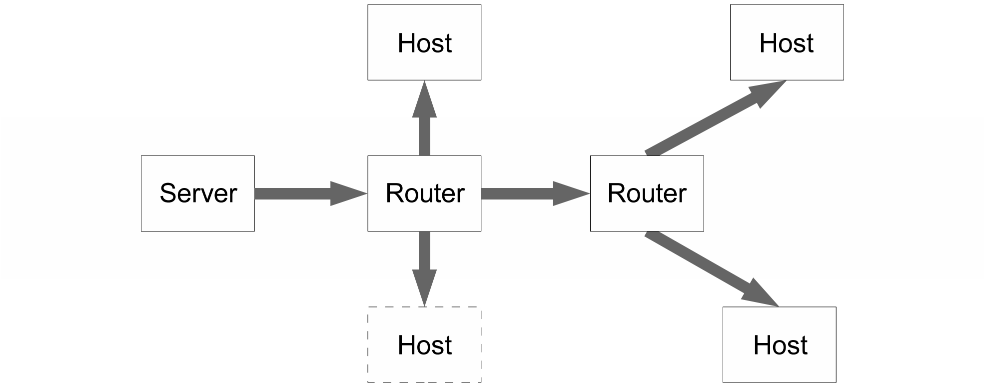


Figure 2.14: Broadcasting packets from one host to all hosts.

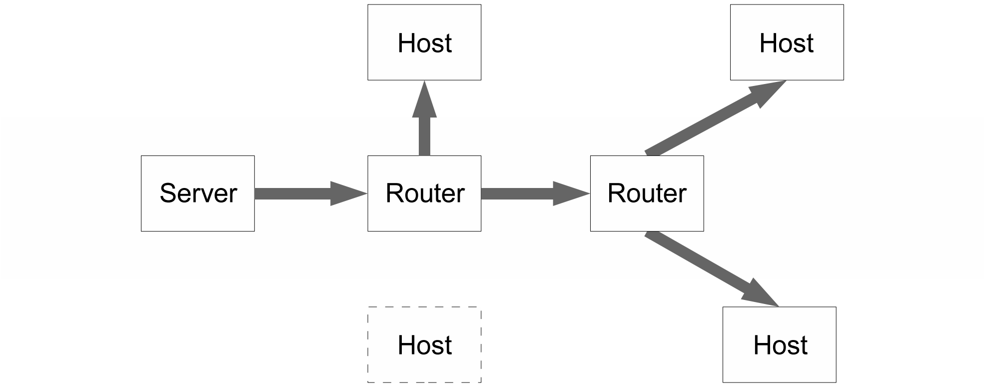


Figure 2.15: Multicasting packets form a server to a group of hosts.

2.5.3 Multicast

Multicast works in a manner similar to that of broadcast for a selected number of hosts. A host subscribes for a stream and becomes a member of that multicast group. Information is then broadcast within the group. A membership inquiry to the multicast group is handled by routers.

Multicast applied in a streaming multimedia application is illustrated in figure 2.15, and the following example. A sender sends information (video) to a multicast address (host group), the receiving hosts (clients) join the group if the video is requested by the client. If a client no longer wants to receive the video carried by the group, the client leaves the group and the multimedia data is no longer forwarded to that client. One example of an application that can benefit from using multicast is an IPTV service by assigning each TV-channel to its own multicast address. A similar approach can be used for an Internet radio application.

	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data link	Host-to-network
1	Physical	

Figure 2.16: OSI and TCP/IP reference models

2.6 Internet Protocol Networks

This section will cover Internet networking protocols in general in addition to Internet Protocol (IP) multicasting. The Internet protocol stack is somewhat different from the seven layer Open Systems Interconnection (OSI) model [35]. The two different approaches are compared in figure 2.16 where it is possible to study the mapping between the two. This thesis only deals with IP networks and the reference to OSI is merely for information.

2.6.1 Internet Protocol

Internet Protocol [36] is located at the network layer of the TCP/IP reference model. The internet protocol ensures that packets arrive at the correct destination and it acts as an interface for a wide plethora of underlying network techniques. The IP version 4 (IPv4) header, illustrated in figure 2.17, contains information about the source host, destination host, header length, total packet length, as well as other additional information.

2.6.2 Transmission Control Protocol

Transmission Control Protocol (TCP) [37] handles connection oriented packet transport and control. The working principle of TCP is well suited to the transport of data where the correctness of the data is prioritized over timely delivery.

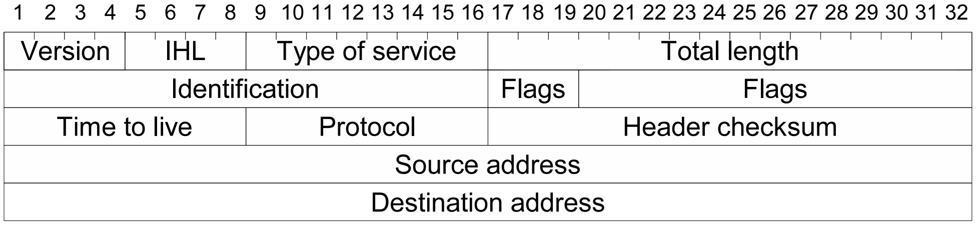


Figure 2.17: IP header fields.

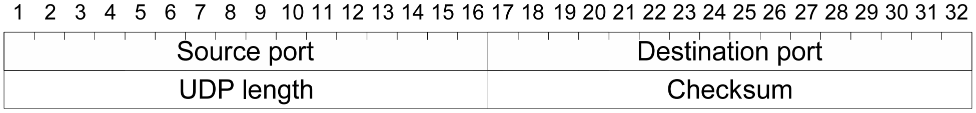


Figure 2.18: UDP header fields.

2.6.3 User Datagram Protocol

Datagrams, or connectionless communication is an efficient method of distributing information with a constrained lifespan. User Datagram Protocol (UDP) is a connectionless transport protocol and all IP packets are handled according to the best effort principle within the network. UDP is often used to transport real-time data, in particular for streaming video. UDP does not have an error detecting capability since there is no specified back-channel defined. This means that if a UDP packet is lost, it is up to the higher layers in the protocol stack to detect and solve the packet-loss situation. However, the UDP header information only specifies the port of the source and destination, packet length and a checksum. To assist in the detection of packet loss or packets out of sequence, the Real-Time Protocol (RTP), Section 2.9.3, has been developed to complement the underlying UDP/IP protocols.

2.6.4 IP Multicasting

Internet Protocol (IP) (Section 2.6.1 multicasting [38][39] specifies how to transport IP datagrams from one host to a group of hosts. This group can consist of zero or some hosts. The group is identified by a particular IP destination address from a dedicated multicast address range. In IPv4 the address range dedicated for multicast [40] is presented in table 2.1. Datagrams sent to a multicast group are forwarded to all the member hosts of that group. However, it is not a requirement that a host is a member of a host group in order to send datagrams to it. Membership is handled by the Internet Group Management Protocol (IGMP) [41].

Table 2.1: IPv4 Multicast address space

Start address	End address	Description
224.000.000.000	224.000.000.255	Local Network Control Block
224.000.001.000	224.000.001.255	Internetwork Control Block
224.000.002.000	224.000.255.000	AD&HOC Block
224.001.000.000	224.001.255.255	ST Multicast Groups
224.002.000.000	224.002.255.255	SDP/SAP Block
224.252.000.000	224.255.255.255	DIS Transient Block
225.000.000.000	231.255.255.255	RESERVED
232.000.000.000	232.255.255.255	Source Specific Multicast Block
233.000.000.000	233.255.255.255	GLOP Block
234.000.000.000	238.255.255.255	RESERVED
239.000.000.000	239.255.255.255	Administratively Scoped Block

2.7 Quality of Service

Quality of Service (QoS) is a very wide concept and is used quite loosely in various situations. The QoS term in a multimedia distribution context often means that resources are reserved and measures are made to guarantee that these reservations are honoured during the session. Real-time services such as IPTV are very sensitive to variations in available bandwidth or excessive delay jitter which makes QoS guarantees particularly important for such services.

2.7.1 Differentiated Services

Differentiated Services (DS) [42] [43] is a method to implement a scalable solution for service differentiation in an IP network. In practice the *type of service*, see figure 2.17, field in the IP header is used to assign the packet to a particular traffic class [44] by a ingress router to the DS-domain. The packet is then distributed through the DS-domain, which contains routers employing a common DS-policy for the routers in the DS-domain. Individual routers act on the class information and can treat packets belonging to different classes differently.

2.8 Video Quality

Video quality assessment can be performed, either using a subjective judgment conducted by a human focus group or by an objective method where the judgment is based on a predefined algorithm.

2.8.1 Subjective Video Quality

The international standard ITU-R BT.500-10 [45] specifies a set of methods which deals with how to set up a subjective test for the assessment of TV pictures. A large number of parameters are specified, such as the audience size, viewing time, and statistical methods to mention but a few. Although the standard is not really applicable to digital video in general it at least provides some information as to how to set up a subjective test. Since subjective methods include a focus group and specialized equipment, they are often too complicated and expensive to use for small investigations under a constrained budget.

2.8.2 Objective Video Quality

Objective methods are ideal for the evaluation of the degradation when the uncontaminated source material is available for reference. An objective picture quality assessment method uses information supplied and a method to derive the metric. The effectiveness of objective methods is often compared with subjective data. Objective methods are often ranked with regards to how much information they require in order to judge a picture and are therefore often classified based on this. Full reference methods use a complete undistorted version of the picture in order to assess it. Peak Signal to Noise Ratio (PSNR) [46] is a popular and widely used full reference objective metric. The Structural Similarity (SSIM) index is a rather new full reference objective method [47] which can be used as a replacement for PSNR in some applications. No-reference methods usually require additional information than that actually provided in the picture to be judged. In this case the assessment method is often specialized to assess a specific type of impairment and is thus rather tailored to the application.

2.9 Multimedia Transport

To transport multimedia information over a packet switched network provides a series of challenges. A multimedia stream generally consists of several individual streams of audio and video. Each one of these individual streams is often referred to as an Elementary Stream (ES) and thus each of the ESs is probably related to others, i.e. representing audio and video from a common capture instance. In this case the timing of the delivery and presentation of the individual ESs is crucial for the perceived quality of the presentation. For example, consider a multimedia stream consisting of one audio ES and one video ES. These two streams have to be delivered and presented in such a fashion that the streams are synchronized in order to achieve the best possible quality. If the video becomes of sync with the audio, this reduces the overall quality impression of the combined multimedia stream even though the individual streams are error free. If the sync skew is below "plus minus" 80ms then the synchronization between the audio and video may be considered to be good [23].

Multimedia transport is also affected by the transmission channel over which the system transmits the signal. Information loss caused by congestion is one plausible event that can cause distortions in the decoded video. For example, if the video content being delivered is compressed using predictive coding there is a risk that a single information loss event can influence several frames. The distortion propagates until a random access point is reached. This effect is also known as *drift*. When large variations in delay occur then this is another factor that can severely impact upon the quality of streamed multimedia transmissions. For example, the data buffer in the receiver can be starved and the decoder can be forced to miss a frame, causing unpleasant jerkiness in the video play-out.

2.9.1 Multimedia Streaming

The term streaming is often used when mentioning transport of multimedia from one host to another. In general, video can either be streamed or downloaded. If multimedia content is downloaded it is received in full before the play-out of the multimedia begins. When multimedia is streamed a portion of the multimedia is forwarded to the receiving host in such a manner that the play-out can begin before the stream has been received in full. The amount of buffer used and the rate of the data received will determine the amount of time before the play-out of the multimedia can begin. In the strictest sense, downloading can be considered to be streaming with a buffer which is sufficiently large to hold the entire feature.

2.9.2 MPEG-2 Transport Stream

MPEG-2 Transport Stream (TS) a protocol for packetizing and multiplexing MPEG streams. TS is a part of the MPEG-2 Systems standard [27] [48].

In practice the audio ES and video ES together with other data are multiplexed into one TS. The TS is a stream consisting of several fixed length packets. Transport stream packets are 188 octets long where four octets comprise a fixed header,as illustrated in figure 2.19. The TS header indicates which ES the payload data belongs to as well as counters for error resilience and fields to indicate scrambling and header extensions. TS is widely used as a transmission technique as specified by the Digital Video Broadcasting (DVB). DVB specifies different ways of transmitting TS over various mediums, including terrestrial (DVB-T), cable (DVB-C), and satellite (DVB-S), to mention the most common.

Since DVB as a organization is heavily linked to the broadcast industry it is natural that it is able to have some influence on the development of the transmission of digital television over networks [49].

2.9.3 Real-time Transport Protocol

The Real-time Transport Protocol (RTP) [50] [51] specifies header extensions for UDP packets to enable transport of real-time data, such as multimedia. The standard

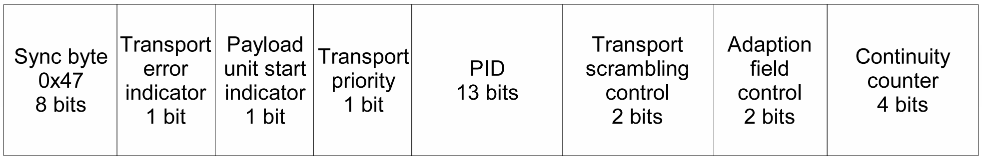


Figure 2.19: Transport Steam packet and header.

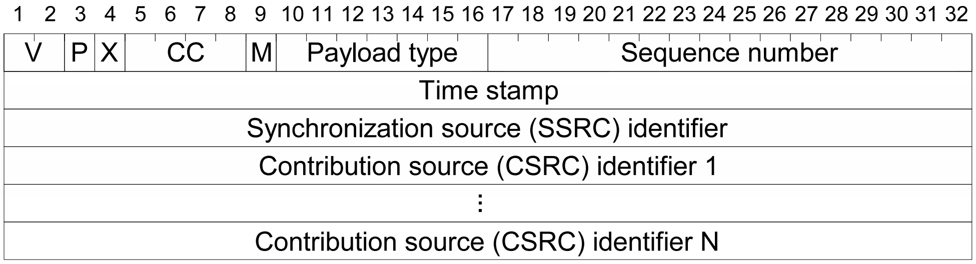


Figure 2.20: RTP header fields.

RTP header format is illustrated in figure 2.20. There are RTP header extensions specified for several video and audio payload formats. For example, H.264[52] and [53]. Since RTP is transmitted using UDP the means for feedback information about the transmission quality is limited. For this purpose the RTP Control Protocol (RTCP) [50] is defined. The primary function of RTCP is to send feedback information about the quality of the RTP distribution.

Given that the recommendation is to use RTP in MPEG2-TS based IPTV distribution [49], it is somewhat surprising that this is not actually used in many cases [54] thus making QoS monitoring impossible.

In general RTP is very well suited for distributing IPTV services. To use Native RTP [54] instead of MPEG2-TS removes unnecessary headers introduced by the MPEG2-Ts. Native RTP transmission also allows for the splitting of streams relating to one program into several streams. For example, a program distributed with MPEG2-TS can consist of one video stream and several audio streams. However, since it is very unlikely that a client would be interested in two simultaneous audio streams, it is a waste of bandwidth to packetetize the program in such a way. By using native RTP for all the streams associated with the program these can be distributed on separate multicast addresses and be received by explicitly joining them. The time-stamp field in the RTP header can be used to synchronize the different streams in the program.

2.10 Internet Protocol Television

Internet Protocol Television (IPTV) has over the recent years gained popularity among the new broadband services. IPTV is a relatively new broadband service, but it is expected to grow rapidly over the next few years [8].

Although IPTV has been mentioned during the last couple of years, some confusion exists around the meaning of the what is or is not IPTV [55]. Even though it appears to be impossible to clearly define ITPV it can actually be very simple. IPTV is TV distributed over IP. However, two services that are often mentioned as being part of the IPTV concept are; linear TV distributed by multicast (live-TV) and Video on Demand (VoD).

The live-TV service is the one considered to be the normal TV service of today, that is, TV-programs are bundled together in a channel following a predefined schedule. In the IPTV deployments over the world at present, live-TV service is distributed by multicast. In general the content is coded using MPEG-2 and multiplexed into a MPEG-2 TS. It is often not necessary to re-encode the video and audio since it they are often delivered to the IPTV playout location (head end) by either satellite or terrestrial transmission and are already coded in MPEG-2. However, since satellite and terrestrial Digital video transmission techniques often bundle several TV-channels together into one TS, it is necessary to re-multiplex the information associated with one TV-channel into a new TS. One TV-channel now resides inside its own MPEG-2 TS and is packetized into UDP/IP packets and sent to the multicast group.

Video on Demand (VoD) is the other of the two mentioned IPTV services. VoD functions as a video rental store accessible through the TV-set. This sounds remarkable, but is actually quite easily implemented in its more basic forms. Several protocols exist which can assist the VoD distribution and examples of these are; RTSP to provide trick play functionality and different content scrambling protocols and conditional access protocols.

IPTV is sometimes confused with Internet-TV [56] which offers low resolution streamed TV content over the Internet, or even sites such as youTube [57]. The border between what can be considered as Internet-TV and IPTV are services including joost [58], babelgum [59], and zattoo [60]. These have a more TV-like user interface and in some cases a more developed and advanced distribution method. However, these services require the user to download and install some proprietary software before the service can be used.

A good comparison between Internet-TV and IPTV is available in an EBU report [61] where factors such as, video quality, resolution, bandwidth, and so on, are listed in order to mark the difference between IPTV and Internet-TV. This thesis uses this definition of IPTV.

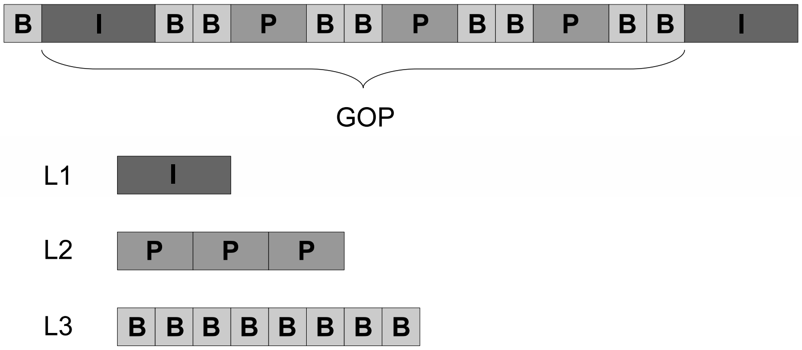


Figure 2.21: Temporal scalability.

2.11 Hierarchical Video Coding

Since the coding of video is based on the removal of redundant information in the video signal it can be argued that some part of the coded information is more important than another. Extending this, it could also be stated that some of the coded information is useless without other information being available. One example of this is that it is impossible to reconstruct a P-frame without access to the reference frame. Hierarchical video coding [16] makes use of this property when ordering the importance of the coded video information. Hierarchical coding of video signals is of great use when examining methods for video transport over capacity constrained channels. Hierarchical video coding makes it possible to reduce the resource requirements of a coded video signal in a graceful way by reducing the resource consumption by dropping the least important information first.

2.11.1 Temporal Scalability

Temporal scalability [16] is one method of organizing the coded video data into different hierarchical layers. In general, temporal scalability is easy to achieve since it is possible to partition a pre-encoded bitstream. One example regarding how temporal scalability can be applied in a GOP video sequence is illustrated in Figure 2.21. In this case the hierarchical structure over one GOP is provided as the I-frame is located in the first layer. The I-frame predicts the following P-frames which constitute the second layer. The third layer consists of all the B-frames in the GOP. One drawback associated with temporal scalability in this manner is that the visual result of the removal of the least important layer drastically reduces the frame update frequency.

2.11.2 Frequency Scalability

Another approach to achieving scalable video encoding is to assign the frequency related information in the picture to different layers. Frequency scalability [16] is

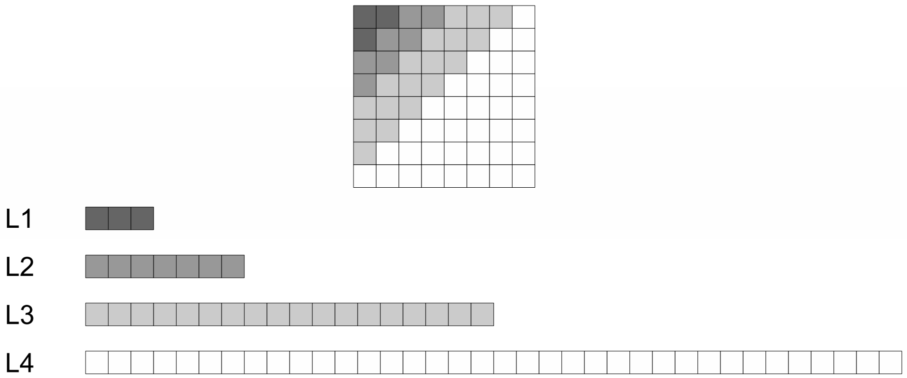


Figure 2.22: Frequency partitioning.

achieved by dividing the transform coefficients into different layers. Low frequency components are assigned to a base layer and higher frequency components are assigned to higher layers. An example of a frequency scalability approach is illustrated in figure 2.22. The result of adding layers to the base layer can be thought of as an increase in the quality.

Chapter 3

Synchronization Frames for Channel Switching

This chapter familiarize the reader with the concept of Synchronization Frames for Channel Switching (SFCS). SFCS was originally designed to work with compressed video but the principles could be applied to all shared information flows subject to temporal redundancy.

3.1 Background

For efficient coding of digital video it is essential to exploit the temporal redundancy in the signal. *Inter* coding, see section 2.2.2, minimizes the temporal redundancy of video sequences and achieves a good compression result. However, *inter* coding introduces dependencies between consecutive frames in a coded video sequence. This dependency is due to the motion- estimation/compensation techniques used. In brief, information in one frame can be reused in other frames. In order to be able to initiate decoding a stream consisting of inter coded frames, intra coded frames are inserted to act as good decoder starting points. The Group of Pictures (GOP) concept specifies an intra coded frame followed by a number of inter coded frames. GOP sizes are often fixed in a sequence. Hence,this will provide suitable stream resynchronization points at a fixed frequency. A stream synchronization point is also called a stream Random Access Point (RAP) [62]. Since *intra* coded frames are by definition larger than inter coded counterparts it makes sense to have as few of them as possible. In figure 3.1 this is illustrated by the rate/distortion plots of H.264 encodings of the 1280x720p 50fps SVT sequence "Fairytale" with the x264 encoder [32] using different GOP sizes. Fewer *intra* frames means better quality or lower bitrates. The only remaining reason for having *intra* coded frames in a stream is to provide RAPs.

The following two sections will address the basics involved in SFCS and also

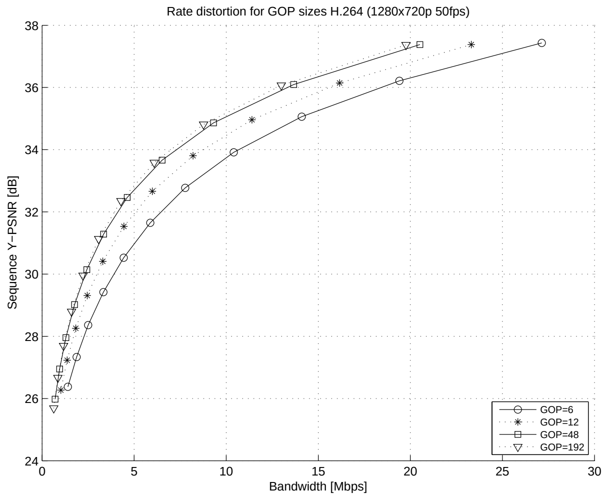


Figure 3.1: Rate distortion plots for the 1280x720p 50fps SVT sequence "Fairytale" at different GOP sizes.

a section covering expressions for bandwidth estimation of SFCS distributed over network links.

3.2 Synchronization Frames for Channel Switching

Synchronization Frame for Channel Switching (SFCS), published in the included papers III-V, is based on the idea that it is redundant to send RAPs at a fixed frequency as is done in GOP based transmission schemes. SFCS separates the RAPs from the *inter* coded stream and transmits them on two separate multicast addresses. Channel changes are performed by first joining the stream with that carrying the RAPs. After receiving one RAP, the RAP stream is left and the *inter* coded stream is joined. The received RAP is spliced with the *inter* coded stream and forwarded to the decoder.

SFCS is designed to function in a streaming environment where several hosts share a common stream originating from one source. A good example is the live-TV service in the IPTV service suite. In such a service TV channels are transmitted from one head-end server to hosts using multicast. One channel is transmitted on one multicast address. When clients want to switch channels, they merely leave the current channel and join the new one. The following example explains how this is done traditionally and is illustrated in figure 3.2 where the transmitted, received, and decoded frames are viewed. The client requests a channel switch from the current channel (1) to channel 2 at time instance 4. The client now immediately leaves the multicast group for channel 1 and joins the multicast group for channel 2. The client starts to receive channel 2 data at time instance 5. The client's decoder inspects the data and waits until a RAP appears in the stream. At time instance 9 the client

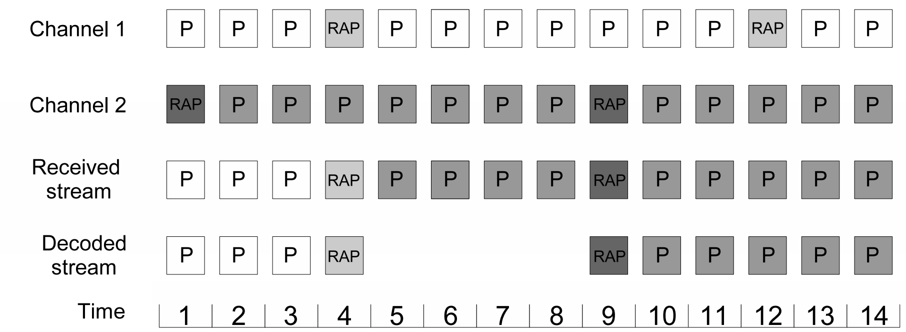


Figure 3.2: Channel switch for a traditional GOP system.

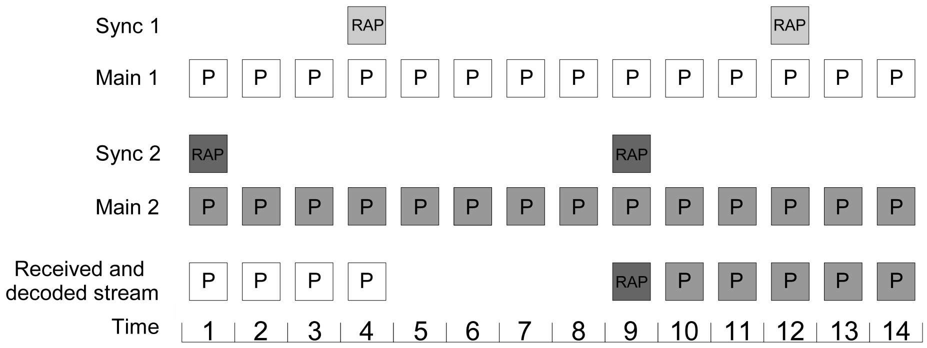


Figure 3.3: Channel switch for a SFCS system.

receives a RAP and can start to decode the stream. The client is synchronized to channel 2 at time instance 9.

The SFCS concept divides the channel into two streams. The steady state stream, which is called the *main* stream, consists of *inter* coded frames only to provide a good compression ratio. The second stream is the synchronization (*sync*) stream, consisting only of RAPs. The RAPs in the *sync* steam are provided at a frequency suitable for the situation. The *sync* stream is only received at the actual synchronization point and is later left. Both the *sync* and *main* are distributed by means of multicast. The following example illustrated in figure 3.3 and explains a SFCS channel switch. The client is synchronized to the *main* stream of channel one and requests a switch to channel 2 at time instance 4. The client leaves channel 1 *main* multicast group and immediately joins the *sync* multicast group of channel 2. The client receives data on the *sync* group at time instance 9, after the whole RAP is received, the channel 2 *sync* multicast group is left. At the same time the *main* group of channel 2 is joined. The client splices the received data from the *sync* stream with the data from the *main* stream and feeds it to the decoder. The client is now synchronized to channel 2.

SFCS can also be used to combat effects of information loss. For example, for some reason the client's decoder looses synchronization with the *main* stream. This

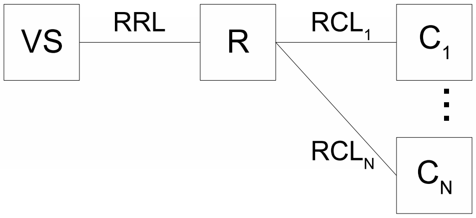


Figure 3.4: The different node and link types in the network.

reason could be that a packet is lost and the decoder is unable to decode the stream correctly. Then the client can join the *sync* stream and resynchronize to the stream. This approach can also be good to use in a packet loss situation which has been caused by congestion. The initial back-off caused by leaving the *main* stream releases resources and therefore resolves the congestion situation.

3.3 Bandwidth Estimation

This section will state bandwidth estimation expressions related to SFCS transmission over multicast enabled networks. There exist three types of nodes on the network, Video Server (VS), Router (R), and clients (C). The server is the origin of all streams. Routers route traffic from the server to other routers or to clients. Clients request streams and are the sink for the streams received. The links between the nodes are categorized into two types. The link which connects one router to another or to the video server is called a Router to Router Link (RRL). A link connecting a router to a client is called a Router Client Link (RCL). A simple network layout with all the components is displayed in figure 3.4.

It is assumed that $N \gg M$, $R_{RG} + R_{PG} = fps$ and $R_{PSM} = fps$. All channels are assumed to have the same RAP/inter-frame size ratio. Clients are assumed to behave equally in terms of issuing synchronization requests (d).

The expected GOP bandwidth in one RCL becomes

$$B_{GRCL} = \left(r_{RG} R_{RG} + R_{PG}\right) P. \tag{3.1}$$

GOP bandwidth estimation expression for the RRL for one channel m evaluates to

$$B_{GRRL,m} = \left(r_{RG} R_{RG} + R_{PG}\right) P. \tag{3.2}$$

And the GOP bandwidth for all the channels (M), assuming $N >> M$ in one RRL the becomes

$$B_{GRRL} = M \left(r_{RG} R_{RG} + R_{PG}\right) P. \tag{3.3}$$

SFCS bandwidth estimation for one RCL

$$B_{SRCL} = \left(R_{PSM} + r_{RSS} R_{RSS} \phi\right) P. \tag{3.4}$$

SFCS bandwidth estimation in a RRL for the channel m

$$B_{SRRL,m} = \left(R_{PSM} + r_{RSS}R_{RSS}\left(1 - e^{-\frac{N_m}{dR_{RSS}}}\right)\right)P. \tag{3.5}$$

For all channels in a RRL the SFCS bandwidth estimation then becomes

$$B_{SRRL} = \left(MR_{PSM} + r_{RSS}R_{RSS}\sum_{m=1}^{M}\left(1 - e^{-\frac{N_m}{dR_{RSS}}}\right)\right)P. \tag{3.6}$$

The bandwidth estimation expressions can be utilized when designing IPTV distribution systems.

Further details of the SFCS concept applied in different environments can be found in the included papers III-V.

Chapter 4

Error Resilience in Video Transmission

Error resilience in video transmission deals with techniques to recover from or resist being affected by information loss. Two aspects of error resilience is addressed in this thesis which are presented in the following two sections.

4.1 Scalable Video

Information loss is sometimes caused by congested routing nodes. Scalable video coding and transmission are based on the idea that it can be beneficial to discard information of low importance first in order to resolve an information loss situation caused by congestion. Receiver-driven layered multicast [63] is one such application.

To understand the impact that a layered video transmission system has on the quality of the decoded video it is important to study the error propagation between the layers.

Layered video coding as well as prioritized packet scheduling are two well known methods that may improve the quality of service level in real-time applications with high bandwidth requirements, and are used over packet switched networks. However, it is often difficult to obtain an idea of, and to quantify, the actual gains that may be achievable, especially from an end-to-end perspective.

4.2 Synchronization Frames for Channel Switching

SFCS (see chapter 3 is a method for a receiving decoder to synchronize to a stream with inter-coded frames only by using a side stream with RAPs. This method can be used to recover from transmission errors by simply issuing a resynchronization to the channel when an information loss situation occurs.

4.3 Contributions

The author has published three papers, I, II, and IV in the field each one addressing different aspects of error resilience related to video distribution over networks.

In paper I, the author present a mathematical model for the estimation of video quality for temporally layered MPEG-2 video subject to information loss. The temporally layered MPEG-2 video divides the frames in one GOP into five different layers. The base layer consists of the I-frame in the GOP. The second layer is the fist P-frame following the I-frame, the third layer consists of the P-frame following the one in layer two. Layer four consists of the last P-frame. Finally, layer five includes all the B-frames in the GOP.

The paper uses the number of distorted macroblocks in the decoded frame as a measure for impairments in a picture. Two types of impairments are recognized in the paper, direct impairments which are the result of information loss in frames belonging to that layer and indirect impairments which are the result of using an impaired frame as a prediction for the current frame. A mathematical expression for the number of expected impairments is formulated. Parameters such as the relationship between the size of the layers and how the amount of information loss in one layer affects other layers are investigated experimentally. The results from the mathematical model are then compared with experimental results and the conformance is found to be good.

Although the investigation is focused on the temporally layered MPEG-2, the proposed methods can be applied to any scalable video transmission system with a hierarchical layer structure.

In paper II the author presents some experimental results obtained from using a temporally layered MPEG-2 video combined with a basic per-layer IP packet prioritization. The goal has been to discover whether a basic scheme is at all useful in combination with this particular source coding method, and if so, how much the objective video quality can be increased during bandwidth constrained periods. The quality is measured in terms of PSNR and the results are compared to the case of equal packet priority. In addition, different packet sizes as well as packet queuing disciplines are used. The conclusion drawn is that using even a relatively simple temporal layering strategy in combination with packet prioritization can quite significantly improve the end-to-end quality of MPEG-2 video, especially in moderately bandwidth constrained situations. Furthermore, packet size and queuing discipline are found to have an impact on decoded video quality. It is interesting to note that the proposed solution shows an improvement in objective quality even for low packet loss ratios.

In paper IV, SFCS is evaluated from an error resilience point of view. In that case the *sync* stream is used to resynchronize the receiver to the channel. In this way the receiver can recover from an information loss situation. Different packet loss ratios are investigated using both the presented analytical model and a software simulation. The effect on the bandwidth in both the RRL and RCL are studied. The conformance between the simulated and the analytical results for moderate frame-loss frequencies are more than satisfactory. The results indicate that SFCS can be used as a method to increase error resilience in live-TV distribution in an IPTV deployment.

Chapter 5

Fast Channel Change

In live-TV distribution over IP it is common to use multicast to distribute the individual TV-channels to the receivers. When a user of the system issues a channel switch it simply leaves the current multicast channel and joins another. The time between the switch to the new channel being issued and it appearing on the screen is denoted as the channel switch time or the channel change time.

The general feeling is that long channel switching times are perceived as annoying. This is also confirmed by a study by Kooij *et al.* [64] who suggests that channel switching times below 0.5 seconds are acceptable, but channel switching times above 0.5 seconds are considered to be annoying.

Fast Channel Change (FCC) describes techniques enabling rapid changes between encodings of different streams. The following example is a good representation of an IPTV channel switch. The client IP-STB receives the channel change order by means of a user request (a user is pressing a button on the remote). The current multimedia stream is left by issuing an IGMP leave message to the nearest router. Immediately the new stream is joined by issuing an IGMP join message to the nearest router. The client IP-STB now awaits packets from the new stream, but depending on the availability of the stream in the router and the multicast routing protocols used this process this can consume a considerable amount of time. However, if the stream is available the waiting time is considered to be small. When packets arrive at the client IP-STB it starts by identifying the stream and searching for a suitable stream position to start the decoding process. This stream position is recognized as a random access point (RAP). A RAP is often an I-frame preceded by information regarding the stream. Compressed audio streams have in general much more frequent RAPs when compared to compressed video. Depending on the RAP distance in the stream the waiting times can range from zero to a considerable amount. When the stream is received, the client IP-STB buffers the steam and initializes the hardware. When the IP-STB is initialized the decoder can start to decode the stream. Depending on the number of bidirectional frames used there can be an additional buffering time before the picture is displayed. To avoid sensitivity to variations caused by the packet arrival rate (jitter) it is wise to build a de-jitter buffer to combat this condition. The fill time of the de-jitter buffer depends on several factors but cannot be ignored when estimating the channel change time.

Fast channel change is a rather wide concept and can include several techniques. How these techniques are implemented depends mostly on the underlying network infrastructure. For example, Cho *et al.* [65] suggests an adjacent groups join method in which it forwards channels neighboring the current channel to the home gateway for more rapid access. Cho *et al.* defines the "channel zapping time" to only include the delays related to the network but does not mention the delays introduced by RAP acquisition or various buffer fill times. Boyce and Tourapis [66] propose a system that utilizes a lower quality stream which is only used in the tune-in moment of the channel. The idea resembles, in many cases, the SFCS system proposed by the thesis author. However, Boyce and Tourapis suggest that a low-quality stream should be multiplexed together with the normal stream upon request. The two streams should then be transmitted and be demultiplexed and spliced in the decoder. This approach requires that stream multiplexing equipment resides near the client, e.g. a DSLAM. Boyce and Tourapis have a patent [67] for the application in a DSL environment. The idea differs from SFCS in the sense that for SFCS the *sync* (low-quality) frames and the *main* (normal) frames in two different streams are distributed on two different multicast addresses.

The ISMA has also addressed rapid channel change in an external review document [62]. In that report the authors provide a more comprehensive view of the FCC area. Some aspects of the document are covered in the next two sections.

5.1 Tune in Channel

One easy means of achieving FCC is to use a tune-in stream (or side stream) during the channel change moment. The tune-in stream consists only of RAPs and is only transmitted on request. One inherent problem associated with the tune-in approach is the decoding drift introduced by the difference between the encodings of the two streams. This approach is further investigated by the author in the included papers III-V.

5.2 Edge Server

Efficient FCC can also be achieved by placing advanced multiplexing and encoding equipment near the clients. A suitable location for such equipment can be a DSLAM or an advanced IP switching equipment. By buffering all the streams that the equipment can offer and having a fast buffer fill by increasing the rate of the stream until the client is synchronized to the stream then waiting until the buffers have reached an acceptable level. This equipment can also serve other purposes, such as a VoD cache. However, this is not addressed by the thesis.

5.3 Contributions

The author has published three papers III-V covering various aspects of the issues using SFCS in a FCC solution.

In paper III the concept of SFCS is first introduced. Network model and bandwidth estimation expressions are formulated. Necessary parameters, such as frame size ratio and channel popularity are identified. A simple test scenario comparing SCFS and GOP is presented. Estimated bandwidth consumption is both calculated from the bandwidth expressions and verified with an experimental software simulation. SFCS is found to have lower bandwidth consumption compared to similar GOP system for the presented scenario.

In paper IV the SFCS concept is further analyzed and adapted to an SI/SP-frames [20] environment. The bandwidth estimation expressions are modified to accommodate SI/SP-frames and is also extended to calculate the effects of information loss, see section 4.3. The paper also targets the inefficiency problem associated with SFCS in the event of excessive number of synchronization requests. The paper suggests an SFCS-GOP hybrid solution to solve this issue. The presented hybrid solution is complete with a design outline and expressions to calculate SFCS-GOP switchover points. Expressions to calculate the expected tune-in time are also presented. The results from the modified bandwidth expressions are compared with the simulation results. Frame size parameters are found by experiment and these are based on encodings of several video sequences. The GOP-SFCS hybrid solution is found to reduce the bandwidth in the RRL by 13% for 10000 connected clients choosing from 250 different channels using Zipf distributed channel popularity.

In paper V the idea of applying SFCS in a GOP-oriented transmission environment is investigated. One problem associated with this approach is that the splice between the *sync* and *main* is difficult to perform without the introduction of prediction mismatch causing drift. The tune-in quality is experimentally investigated by slicing an MPEG-2 sequence and recording the quality in terms of luminance PSNR. The mean quality difference is as low as 1.6 dB and is judged to be acceptable. Modified bandwidth estimation expressions are presented. An MPEG-2 encoding is used as a source for frame size parameters. Two different approaches for the encoding of the *sync* stream are investigated. The quality difference between the two encoding approaches is very small and probably needs to be investigated further. SFCS is found to reduce bandwidth with 9% compared to a similar GOP system for the given parameters.

Chapter 6

Summary

Over recent years the development of the Internet and services distributed over Internet have grown rapidly. Many households gain access to the Internet through broadband networks. In addition to offering a high bandwidth connection to pure Internet services, such as, WWW, e-mail, and Internet-TV it offers a new set of services including IP-telephony and IPTV. A bundle of these services is often referred to as triple-play service. The most resource demanding part of the triple-play bundle is the IPTV distribution. IPTV consists of several techniques for transmitting TV services over IP networks. It includes Video on Demand, Pay-per-view, but also linear TV programs (live-TV). Live-TV is often distributed using multicast and uses complex schemes for compression of the multimedia signal. Compression of multimedia often relies on techniques that reuse known information to increase the compression efficiency. This makes it extra sensitive to information loss. Hence, IPTV is a service both demanding high bandwidth as well as a near loss free transmission environment.

This thesis addresses two areas related to IPTV distribution, error resilience and fast channel change. The two following sections summarize the contributions in the two fields.

6.1 Error Resilience in Video Transmission

The thesis addresses three areas in the error resilience field. The first area deals with the understanding of error propagation in layered video and the implications of information loss on decoded video quality. The author has contributed by means of the development of a mathematical model to estimate video quality in layered video. The mathematical model is verified by an experiment. The method and results are presented in paper I.

The second area addresses the usefulness of the combined effect of queuing strategies and priority dropping of scalable video and the results are presented in paper

II.

The third area consist of a study using tune-in streams to recover from decoder failure due to information loss. The author has addressed this area in paper IV.

The details regarding the methods and the contributions made by the author are presented in chapter 3, chapter 4 and in papers I, II and IV.

6.2 Fast Channel Change

Fast Channel Change aims to reduce the time it takes from issuing a channel change to the new channel appearing on the screen. In IPTV FCC is applied to the live-TV distribution. The author has proposed a novel idea regarding the use of side stream tune-in information to speed up channel changes and reduce bandwidth consumption. Detailed information about the methods and the contributions by the author can be found in chapter 3 and 5. The author's work in the area is presented in paper III, IV, and V.

SFCS has also successfully being implemented in software IP-STB using C/C++ and open source software as part of a master student thesis work [68]. The results in terms of bandwidth consumption, tune-in time, and tune-in quality are in line with the theoretical numbers presented in the included papers.

6.3 Future Work

This section states possible future work related to the field of IPTV distribution.

6.3.1 Error Resilience Coding

The use of scalable video coding techniques has been hindered by the lack of efficient scalable video coding techniques. With the introduction of Amendment 3 in the MPEG-4 AVC, there are promises [69] of scalable video coding performances equal to single layer coding. This should be of interest to researchers and encourage them to revisit the area of scalable coding and transmission. While the consumer market for new TV-sets is dominated by HDTV capable equipment, a great deal of equipment still exists which has already been installed in homes and which is not HDTV compatible. It is also possible to envisage a growth in interest for video being distributed to small devices such as mobile phones.

Scalable video coding techniques are of great interest for this heterogeneous receiver environment. It is possible to imagine a solution where it is easy to use parts of the same signal to feed equipment with different demands on the resolution of the signal. The results in paper I can also be modified to such a system. The error resilience capability of SFCS enabled systems could be investigated further. Studies focusing on the effects on video quality could be interesting to conduct.

6.3.2 Fast Channel Change

The introduction of H.264 as the future video coding format for IPTV makes it possible to code the material with large intra frame distances without the risk of introducing errors. The increased intra frame distance does have a downside. When attempting to decode such a stream the decoder has to discard all the frames until an intra coded frame arrives. Fast channel change techniques address this problem.

FCC is still quite a new research area and there is still much research requiring to be done. The most promising approaches are probably those dealing with different methods for tune-in streams and edge-server solutions. Since the development and deployment of IPTV systems currently drives the research of FCC it is reasonably, in parallel, to study the field of IPTV services in general.

6.3.3 IPTV

IPTV is still in its infancy, but the rapid growth will boost research efforts in areas concerning IPTV distribution and IPTV services. In this section the author will give some suggestions for areas suitable for future research.

IPTV Standardization

Although several attempts to standardize IPTV are ongoing and there are numerous standardization bodies addressing this issue, IPTV is still somewhat strangely not sufficiently standardized. For IPTV to become the success everyone hopes, there must be more work conducted within the standardization sector.

Future of IPTV Distribution

People watch programs, not channels. Bearing this in mind, it appears rather strange to packetize programs in sequence over time to suit one set of connected users of the system. This solution will always be suboptimal from the single individual perspective. A better solution would to tailor the available media individually to each user of the system. The conclusion that can be drawn from that statement is that the current distribution model for traditional time-table channel based IPTV transmission is over. Future IPTV distribution must be VoD based, at least for the majority of the distributed programs.

In this case the number of simultaneous VoD sessions will place the network and distribution equipment under considerable stress. Considering a unicast solution to the VoD problem does not add up and although this has been mentioned before it is very easy to illustrate by means of a simple example. A network serving 100000 customers with individual VoD sessions where each session will load the network with 10 Mbps (HDTV quality). All together this would load the video server with 1 Tbps, which can be considered to be quite much. There is also a high probability that two or more customers are interested in the same content at approximately the same time. This example emphasize the necessity for efficient VoD transmission techniques.

Bibliography

[1] Post och Telestyrelsen, “Bredband i Sverige 2005,” Post och Telestyrelsen, Tech. Rep. PTS-ER-2005:24, june 2005.

[2] ——, “Bredband i Sverige 2006,” Post och Telestyrelsen, Tech. Rep. PTS-ER-2006:22, june 2006.

[3] ——, “Bredband i Sverige 2007,” Post och Telestyrelsen, Tech. Rep. PTS-ER-2007:17, june 2007.

[4] S. Appleby, B. Crabtree, R. Jeffery, P. Mulroy, and M. Nilsson, “Video coding and delivery challenges for next generation iptv,” *BT Technology Journal*, vol. 24, no. 2, pp. 174–179, 2006.

[5] P. Pfeffer, “IPTV: Technology and development predictions,” *Fiber and Integrated Optics*, vol. 25, no. 5, pp. 325–246, September-October 2006.

[6] G. Held, *Understanding IPTV*. Auerbach Publications, 2007.

[7] J. Weber and T. Newberry, *IPTV Crash Course*. Mc Graw Hill, 2007.

[8] Informa Telecoms & Media, “IPTV: a global analysis (2nd edition),” Informa Telecoms & Media, Tech. Rep., sep 2006.

[9] G. Andersson. (2007, May) Iptv - teknikskifte i klass med mobiltelefonin. Elektroniktidningen.

[10] (2007, November 2) Telia Sonera. [Online]. Available: http://www.telia.se/

[11] (2007, November 2) Bredbandsbolaget. [Online]. Available: http://www.bredband.com/

[12] K. Kerpez, D. Waring, G. Lapiotis, J. B. Lyles, and R. Vaidyanathan, “IPTV service assurance,” *Communications Magazine, IEEE*, vol. 44, no. 9, pp. 166–172, September 2006.

[13] K. Imran, M. Mellia, and M. Meo, “Measurements of Multicast Television over IP,” in *15th IEEE Workshop on Local & Metropolitan Area Networks, 2007. LANMAN 2007.*, jul "10–13," 2007.

[14] EBU Project Group B/MAE, "EBU Evaluations of Multichannel Audio Codecs," The European Broadcasting Union, Tech. Rep. TECH 3324, September 2007.

[15] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. Chapman & Hall, 1997.

[16] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, New Jersey, USA: Prentice-Hall, Inc., 2002.

[17] *FUTURE HIGH DEFINITION TELEVISION SYSTEMS: television production equipment for a progressively scanned image format of 1920 horizontal by 1080 vertical resolution at 50 and 60 Hz frame rates*, EBU EBU Statement EBU R115-2005, may 2005.

[18] Joint Photographic Experts Group. (2007, October 19) http://www.jpeg.org.

[19] R. D. Kell, "Improvements Realting to Electric Picture Transmission Systems," U.S. Patent 341 811, may 25, 1929.

[20] M. Karczewicz and R. Kurceren, "The sp- and si-frames design for h.264/avc," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 637–644, July 2003.

[21] N. Färber and B. Girod, "Robust h.263 compatible video transmission for mobile access to video servers," in *Proceedings of International Conference on Image Processing*, vol. 2, October 1997, pp. 73–76.

[22] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd, 2003.

[23] Z.-N. Li and M. S. Drew, *Fundamentals of Multimedia*. Pearson Educational International, 2004.

[24] *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s*, International Organization for Standardization Std. ISO/IEC 11 172, 1993.

[25] D. L. Gall, "Mpeg: a video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[26] *Information technology - Generic coding of moving pictures and associated audio information*, International Organization for Standardization Std. ISO/IEC 13 818, 1994.

[27] *Information Technology - Generic Coding of Moving Pictures and Associated Audio: Systems Rec. H.222.0*, International Telecommunication Union Std. ITU-T Rec. H.222.0, Rev. E, November 1994.

[28] *Information Technology - Generic Coding of Moving Pictures and Associated Audio: Video Rec. H.262*, International Telecommunication Union Std. ITU-T Rec. H.262, 1996.

[29] *Information technology - Coding of audio-visual objects*, International Organization for Standardization Std. ISO/IEC 14 496, 1999.

[30] (2007, November 19) Divx video codec. [Online]. Available: www.divx.com

[31] *Advanced video coding for generic audiovisual services*, International Telecommunication Union Std. ITU-T Rec. H.264, Rev. (03/2005), 2005.

[32] (2007, December 10) x264 - a free h264/avc encoder. [Online]. Available: http://www.videolan.org/developers/x264.html

[33] (2007, June 5) FFmpeg Download and SVN. [Online]. Available: http://ffmpeg.mplayerhq.hu/download.html

[34] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, July 2003.

[35] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Prentice-Hall, 1996.

[36] J. Postel. (1981, September) RFC 791 - INTERNET PROTOCOL. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc0791.txt

[37] ——. (1980, January) RFC 761 - TRANSMISSION CONTROL PROTOCOL. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc0761.txt

[38] S. E. Deering and D. R. Cheriton. (1985, December) RFC 966 - Host Groups: A Multicast Extension to the Internet Protocol. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc0966.txt

[39] S. Deering. (1989, August) RCF 1112 - Host Extensions for IP Multicasting. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc1112.txt

[40] Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper. (2001, August) RFC 3171 - IANA Guidelines for IPv4 Multicast Address Assignments. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc3171.txt

[41] W. Fenner. (1997, November) RFC 2236 - Internet Group Management Protocol, Version 2. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc2236.txt

[42] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. (1998, December) RFC2475 - An Architecture for Differentiated Services. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[43] F. Halsall, *Computer Networkning and the Internet*, 5th ed. Pearson Education Limited, 2005.

[44] K. Nichols, S. Blake, F. Baker, and D. Black. (1998, December) RFC2474 - Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc2474.txt

[45] *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union Std. BT.500-10, 2000.

[46] *Digital Transport of One-Way Video Signals - Parameters for Objective Performance Assessment*, American National Standards Institute Std. T1.801.03-1996, February 1996.

[47] Z. Wang, A. C. B. amd Hamid Rahim Sheikh, and R. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[48] W. Fischer, *Digital Television A Practical Guide for Engineers*. Springer, 2004.

[49] *Digital Video Broadcasting (DVB); Transport of DVB Services over IP*, Digital Video Broadcasting (DVB) Std. DVB Document A086, July 2004.

[50] H. Schulzrinne, S. Casner, R. Frederickand, and V. Jacobsson. (1996, January) RFC 3550 - RTP: A Transport Protocol for Real-Time Applications. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[51] 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, March 2001.

[52] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer. (2005, February) RFC 3984 - RTP Payload Format for H.264 Video. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc3984.txt

[53] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata. (2000, November) RFC 3016 - RTP Payload Format for MPEG-4 Audio/Visual Streams. The Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/rfc/rfc3016.txt

[54] A. MacAulay, B. Felts, and Y. Fisher, "IP streaming of MPEG-4: native RTP vs MPEG-2 transport stream," envivio, Tech. Rep., October 2005.

[55] L. Ellis, "IPTV: Still Too New To Define," *Multichannel News*, no. 25, June 2006.

[56] (2007, November 23) Beeline TV. [Online]. Available: http://www.beelinetv.com/

[57] (2007, November 7) YouTube. [Online]. Available: www.youtube.com

[58] Joost. (2007, October 19) http://www.joost.com.

[59] (2007, November 23) Babelgum. [Online]. Available: http://www.babelgum.com/

[60] (2007, November 23) Zattoo - TV to Go. [Online]. Available: http://zattoo.com/

[61] F. Kozamernik and L. Vermaele, "Will broadband tv shape the future of broadcasting?" EBU Technical Review, Tech. Rep., April 2005.

[62] D. Singer, N. Färber, Y. Fisher, and J. Fleury, "Fast channel changing in rtp," INTERNET STREAMING MEDIA ALLIANCE ISMA, Tech. Rep., 2006.

[63] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 1996, pp. 117–130.

[64] R. Kooij, K. Ahmed, and K. Brunnström, "Percived quality of channel zapping," in *Proceedings of th efith IAESTED International Conference Communication Systems and Networks*, August 28-30 2006, pp. 155–158.

[65] C. Cho, I. Han, Y. Jun, and H. Lee, "Improvement of channel zapping time in iptv services using the adjacent groups join-leave method," in *The 6th International Conference on Advanced Communication Technology*, vol. 2, 2004, pp. 971–975.

[66] J. M. Boyce and A. M. Tourapis, "Fast efficient channel change [set-top box applications]," in *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, jan 2005, pp. 1–2.

[67] J. Boyce and A. Tourapis, "Method and apparatus enabling fast channel change for dsl system," World Intellectual Property Organization Patent WO/2005/112 465, November 24, 2005.

[68] Y. Li, "Native RTP fast channel change for a software IP-STB," Master's thesis, Mid Sweden University, 2007.

[69] (2007, July) MPEG Debuts Scalable Video Coding. SC29WG11 Press Release 31. [Online]. Available: http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/WhatsNew/SC29WG11PressRelease31.pdf

Biography

Ulf Jennehag was born on the 26th of November 1972 in Sundsvall, Sweden. He received his Master of Science in Electrical Engineering from Mid Sweden University, Sweden in January 2000. In 2000 he joined the Multimedia and Communications research (MUCOM) group in Mid Sweden University as a Ph.D student. He received the degree of Licentiate of Technology in Teleinformatics from Royal Institute of Technology (KTH) in 2005. Jennehag has been employed as a development engineer at Acreo AB during six months in 2007.

Included Papers

Paper I

Numerical Modeling of Transmission Errors and Video Quality of MPEG-2

Tingting Zhang, Ulf Jennehag, and Youshi Xu

Published in
Signal Processing: Image Communication,Volume 16, Issue 8, Pages 817-825,
May 2001.

ELSEVIER

Signal Processing: *Image Communication* 16 (2001) 817–825

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Numerical modeling of transmission errors and video quality of MPEG-2

Tingting Zhang*, Ulf Jennehag, Youshi Xu

Mid-Sweden University, S-851 70 Sundsvall, Sweden

Abstract

To efficiently combat the signal loss of MPEG-2 transmission over unreliable networks, priority encoding transmission, unequal packet loss protection and priority dropping techniques have been studied in many papers. Those studies are based on the qualitative analysis of different importance of signals, without quantitative investigation of signal loss effect on video quality. In this paper, MPEG-2 packet loss effect on video quality is quantitatively investigated, a temporal layered signal model is described and evaluated, a quality measure for reconstructed pictures called macroblock impairment ratio is suggested and defined. The investigation and the model are specified for MPEG-2, but the principles and the methods are suitable for any layered video. These are useful for the development of efficient schemes and protocols for packet video transmission over unreliable networks. © 2001 Published by Elsevier Science B.V.

Keywords: MPEG-2; Packet loss; Video quality

1. Introduction

Digital audio and video signals integrated with computers, telecommunication networks, and consumer products have resulted in the information revolution. To have available digital compression methods for entertainment TV for different transmission media, such as satellite, cassette tape, over-the-air and cable TV, the International Organization for Standardization (ISO) has developed a standard known colloquially as MPEG-2 and officially as ISO 13818 [7]. Since then, a range of digital video broadcast (DVB) services have appeared and MPEG-2 standard has been employed. The near future is likely to witness the emergence of MPEG-2-based video applications for business and ultimately private mobile users, indoor wireless users and Internet, to be compatible with cable-, terrestrial- and satellite-based DVB and MPEG-2 video databases. For example, experimental mobile video communication which employs MPEG-2 has been tested in Europe [10].

Distribution of video data in the more hostile propagation environment of terrestrial radio networks, mobile networks and Internet, frequently suffers interference, multipath fading, temporal shadowing and network congestion resulting in video packet loss. As is well known, MPEG-2 has three main types of pictures [7,8]: intra-pictures (I), predictive pictures (P) and bi-directional predictive

---

* Corresponding author.

E-mail address: Tingting.zhang@ite.mh.se (T. Zhang).

PII: S0923-5965(01)00009-1

pictures (B). The I-pictures are intra-coded without reference to other pictures. Between I-pictures are a number of forward-predicted P-pictures. A P-picture is coded with reference to the previous I- or P-picture. Between I–P/P–P/P–I pictures are a number of B-pictures. A B-picture is coded with reference to the two I–P (P–P, P–I) pictures. As a result, a packet loss in different types of coded pictures due to noisy channels or congested networks will impair different numbers of reconstructed pictures. This implies that data packets in different types of coded pictures (I, P, B) have different importance for the reconstruction of pictures in the decoders.

With regard to the different importance of packets in I, P and B coded pictures, priority encoding transmission (PET) and unequal packet loss protection (UPLP) as well as priority dropping have been studied in a number of papers, for example in [1,3,5,6,9,12]. These methods will more efficiently utilize the bandwidth and provide an acceptable video quality service. However, these studies are based on the qualitative analysis of different importance of packets, without quantitative investigation of packet loss effect on video quality.

In this paper we intend to investigate the effect of packet loss in different coded pictures (I, P, B) on the video quality and to provide a statistical numerical model for the design of MPEG-2 video transmission in lossy channels or congestion networks. To measure the video distortion caused by packet losses due to unreliable transmission media, a criterion called macroblock impairment ratio (MBIR) is proposed and defined.

The paper is organized as follows. A numerical temporal layered model for MPEG-2 is given in the next section. Then in Section 3, a new measure of video transmission quality, MBIR (macroblock impairment ratio) is suggested and defined. In Section 4, a mathematical model for video transmission with priority protection or priority dropping is described and the MBIR of reconstructed video pictures is mathematically derived and experimentally evaluated. Finally, Section 5 presents conclusions and discussions.

2. A temporal layered model for MPEG-2

In MPEG-2, a sequence of transmitted video pictures is divided into a series of group of pictures

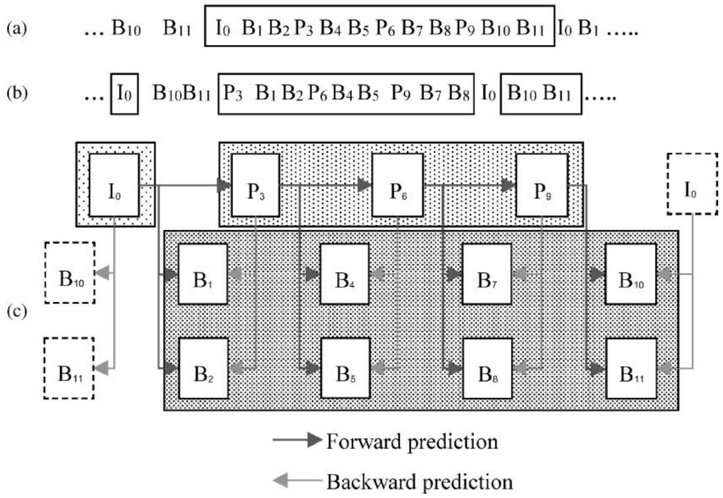


Fig. 1. A typical GOP construction in MPEG-2 coding, with an I-picture every M pictures and a P-picture every N pictures ($M = 12$, $N = 3$). (a) A group of pictures in presentation order, (b) pictures in transmission and coding order, and (c) the predictions in a GOP.

(GOPs), where each GOP begins with an I-picture followed by an arrrangement of P-pictures and pictures. A typical GOP construction, which is recommended and widely adopted in many MPEG-2 products, is shown in Fig. 1. Our study is based on the typical GOP construction with $M = 12$ and $N = 3$.

The compression in I-picture takes each individual picture and treats it in isolation from all others. The most common algorithms are based on the discrete cosine transform (DCT). The image is divided into macroblocks. Each macroblock contains several DCT blocks which are chroma and luminance components despite the different pixel spacing. As an intra-coded picture, the I-picture can be decoded in isolation, which requires no data from other pictures in order to be reconstructed.

A P-picture is constructed in the decoder by taking the previous I- or P-picture and shifting it according to the motion vector transmitted for each macroblock. B-pictures are reconstructed bidirectionally using data from the I- or P-pictures before and after.

If a packet in a coded B-picture data called B-packet is lost, it influences the quality of that picture only. If a P-packet is lost, it influences not only the current picture, but also the consecutively transmitted pictures in the GOP. If an I-packet is lost, all the pictures in the GOP as well as B-pictures at the end of the previous GOP are influenced.

To reflect the inheritance relationship of impairment between I-, P- and B-pictures in a GOP, we can simply and qualitatively model the MPEG-2 signal in five layers as shown in Fig. 2. In this instance, MPEG-2 can be interpreted in such a way that the input signal is compressed into five discrete layers and arranged in a hierarchy that provides temporal progressive refinement, and an upper layer is a temporal enhancement to lower layers.

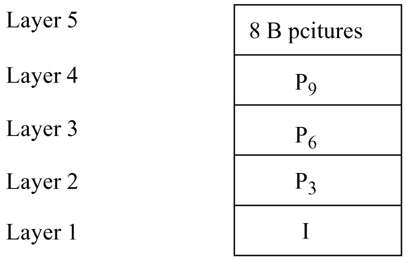


Fig. 2. A temporal layered model for typical MPEG-2 GOPs.

In general, let a layered video code have n layers: layer 1 is the base layer and layer n is the uppermost layer. An n-layered video can be modeled as a vector [12].

$$\bar{D} = (D_1, D_2, \ldots, D_n), \tag{1}$$

where D_i is the relative data rate normalized by the average data rate D_{av} of the video streams, and we have

$$\sum_{i=1}^{n} D_i = 1. \tag{2}$$

Specifically, a typical MPEG-2 signal in Fig. 2 can be modeled as

$$\bar{D}_{\text{MPEG-2}} = (D_1, D_2, D_3, D_4, D_5). \tag{3}$$

We are now interested in a general numerical representation of the model. For this reason, we have investigated video clips of 6 different kinds of MPEG-2 programs, called BBC Sequence 3, Cactus and Comb, Flowers, Mobile and Calendar, Susie, and Table Tennis. These clips are available in the bit-rates of 4, 6, 8, 12 and 18 Mbps from the Tektronix ftp server [11]. The normalized data rates D_i and the standard deviations σ_i ($i = 1, \ldots, 5$) for different data rates ($D_{\mathrm{av}} = 6, 8, 12, 18$ Mbits), as well as the mean values over all the above data rates are shown in Table 1. Since we have found that layers 2, 3 and 4 have the same rate and deviation, the total rate and deviation of the three layers are listed in the table. From the table, we can find that (a) the standard deviation is much lower than the normalized rate D_i, usually $\sigma_i < 0.01 D_i$. (b) The normalized data rate, D_i, is somewhat related to the average data rate (D_{av}). However, the relative difference between any and the mean value is less than 15%. Therefore, we can take the average values and numerically model MPEG-2 signal as

$$\bar{D}_{\text{MPEG-2}} = (0.27, 0.11, 0.11, 0.12, 0.39). \tag{4}$$

Table 1
Normalized data rate and standard deviation on five layers for video programs with data rate = 4, 6, 8, 12, 18 Mbits/s

D_{av} (Mbits/s)	Layer	D_i	σ_i
4	L_1	0.30999	0.00202
	$L_2 + L_3 + L_4$	0.34866	0.00043
	L_5	0.33902	0.00136
6	L_1	0.28211	0.00234
	$L_2 + L_3 + L_4$	0.35405	0.00758
	L_5	0.36134	0.00197
8	L_1	0.26586	0.00295
	$L_2 + L_3 + L_4$	0.34177	0.00072
	L_5	0.39018	0.00150
12	L_1	0.22937	0.00122
	$L_2 + L_3 + L_4$	0.33055	0.00044
	L_5	0.43769	0.00088
16	L_1	0.22937	0.00122
	$L_2 + L_3 + L_4$	0.33055	0.00044
	L_5	0.43769	0.00088
Mean	L_1	0.26334	0.00275
	$L_2 + L_3 + L_4$	0.34111	0.00056
	L_5	0.39319	0.00277

This model is useful for (a) the choice of packet length, (b) layered receiving and bandwidth assignment, (c) developing more accurate models, (d) the design of priority encoding transmission (PET), unequal packet loss protection (UPLP) and priority dropping, and (e) estimating the quality of reconstructed pictures.

3. Macroblock impairment ratio as a quality measure

To reflect the effect of video transmission over lossy channels on the video quality, we now introduce the concept of MBIR as a criterion for measuring video transmission quality and for rating the performance of unequal packet loss protection, priority encoding transmission and priority dropping techniques. There is a wealth of literature on the problem of finding a good objective metric that reflects perceptual distance as induced by the human visual system, but the most commonly cited metric is the so-called peak signal-to-noise ratio (PSNR) and has been adopted in some video standards [2]. The definition of PSNR is

$$\text{PSNR} = 10\log_{10}\left(\frac{V_{\text{peak}}^2}{\dfrac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}[V_0(i,j) - V_r(i,j)]^2}\right), \tag{5}$$

where the peak-to-peak value of sampled video signal, which determines the maximum luminance value in the allowed voltage swing, is the original signal at pixel (i,j), $V_r(i,j)$ is the reconstructed signal, $N_1 \times N_2$ is the picture size.

This metric, PSNR, is useful in applications such as image coding, where the encoding process introduces degradation in the output image, almost everywhere in the image. But, the packet-lossy channel introduces impairment in some areas of reconstructed picture. For example, in Fig. 3 pictures (b) and (c) have the same PSNR value of 20 dB over the whole picture. Since small random errors for instance quantization noise are distributed in the whole picture (b), the picture is noisy, but seems acceptable. For picture (c), since large errors resulting from packet loss are distributed in small areas of the picture, the distortion is definitely unacceptable. Therefore, a specific criterion is needed for the specific problem of packet loss effect on the video quality.

A picture in an MPEG-2 stream contains a number of slices which in turn contain a variable number of macroblocks (MB). A 4:2:0 MB contains the coefficients from six DCT blocks, four Y and two chroma (C_b and C_r), whereas a 4:2:2 MB contains four Y and four chroma (two C_b and two C_r). The motion vectors for each MB are added to form a compressed P-picture, the B-picture data are selected on an MB basis. Thus a lost packet will damage at least one MB in a picture. In other words, an MB is a unit of impairment. In the experiments we have found that:

1. If the PSNR at an MB is less than 30 dB, the impairment is definitely noticeable, otherwise it is not so obvious.
2. If the image data of an MB is erased and replaced with random noise, the MB PSNR is almost always less than 30 dB.

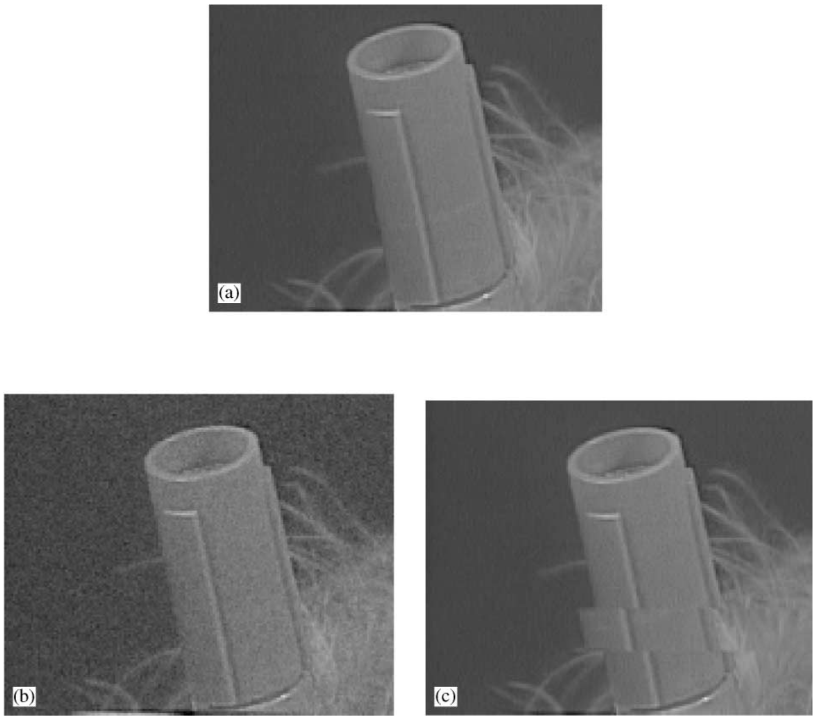


Fig. 3. Reconstructed pictures (partial) with the same PSNR = 20 dB. (a) Original picture, (b) distortion from random noise, and (c) distortion from packet loss.

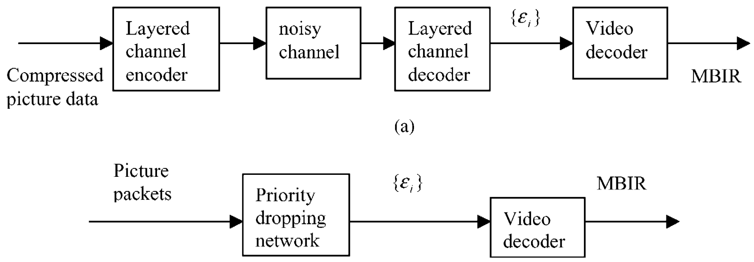
In our experiment, we define the MB impairment as its PSNR less than 30 dB. To measure the quality of reconstructed picture with impairments due to noisy channels or congestion networks, we define the MB impairment ratio (MBIR) as the probability that an MB is impaired, that is

$$\mathrm{MBIR} = P_r[\text{a macroblock is impaired}]. \qquad (6)$$

4. Modeling of unequal protection and priority dropping

Fig. 4 illustrates the block diagrams for (a) PET or UPLP and (b) priority dropping systems, in which ε_i is the packet loss probability in the ith layer. If ε_i has the same value for different layers, they are equal protection or equal dropping systems. How to design the unequal protection codes or priority dropping protocols is beyond the scope of the paper. We are now interested in how to find the MBIR if the set of packet loss probabilities ε_i is given, that is, how the packet losses affect picture quality. As a result, this investigation will provide the preliminary knowledge for the design of channel codes and network protocols.

The model in Eq. (1) does not accurately describe the inherent relationship of impairment between the layers. A lost B-packet damages more MB in the B-picture than a lost P-packet does in the P-picture itself, and much more than an I-packet does in the I-picture, because a coded B-picture needs a smaller number of bits to represent than a P-picture, and much less than an I-picture. On



(a)

(b)

Fig. 4. Block diagrams for (a) priority encoding transmission (PET) or unequal packet loss protection (UPLP), (b) priority dropping systems.

the other hand, errors can propagate from a low layer to higher layers. In addition, not only does the error propagate temporally but it also spreads spatially due to motion compensated prediction [4].

There are two types of impairments: the direct and the indirect ones. The direct impairments on a frame are caused by lost packets of this frame. The indirect impairments are introduced by the reference pictures with impairments. It can be observed that the direct and indirect impairments are sometimes overlapped, that is, the reference data from lower layers are impaired and the data of the current MBs are also impaired. We define the self-induced impairments as the part of direct impairments, which are not overlapped with indirect impairments.

In the layered video, an MB impairment (shortly called an impairment) at lower layers will propagate to the upper layers. To reflect the propagation, an n-layered video can further be modeled by a matrix, called MB impairment propagation matrix (MIPM) and denoted $\bar{E}$, that is,

$$\bar{E} = [E_{ij}]_{i,j=0,1,\dots,n}, \tag{7}$$

in which E_{ij} means that if there is one self-induced impairment at layer i this impairment will result in E_{ij} indirect impairments at layer j.

By computer investigation of the above six MPEG-2 video clips, we find that the MIPN of MPEG-2 can be represented as follows:

$$\bar{E}_{\text{MPEG-2}} = \begin{bmatrix} 1.00 & 1.47 & 1.63 & 1.72 & 13.57 \\ 0 & 1.00 & 1.53 & 1.75 & 10.73 \\ 0 & 0 & 1.00 & 1.52 & 7.25 \\ 0 & 0 & 0 & 1.00 & 3.91 \\ 0 & 0 & 0 & 0 & 1.00 \end{bmatrix}. \tag{8}$$

We now specify the investigation for MPEG-2, but the ideas and methods are suitable for any layered video. Denote the average number of self-induced impaired macroblocks in different layers at a GOP as a vector,

$$\bar{W} = \{W_i\} = (W_1, W_2, W_3, W_4, W_5), \tag{9}$$

and the average number of impairments in the layers,

$$\bar{\lambda} = \{\lambda_i\} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5), \tag{10}$$

which include both self-induced and indirect impairments. Then we have

$$\bar{\lambda} = \bar{W}\bar{E}. \tag{11}$$

Let N_{pic} be the number of pictures in a GOP, $N_{\text{pic},i}$ the number of pictures at the ith layer in a GOP and N_{mk} the number of MBs in a picture. The MB impairment rate at the ith layer is

$$(\mathrm{MBIR})_i = \frac{\lambda_i}{N_{\mathrm{pic},i} N_{\mathrm{mk}}}. \tag{12}$$

From the MPEG-2 model in Fig. 1, we have $N_{\mathrm{pic},1} = \cdots = N_{\mathrm{pic},4} = 1$, $N_{\mathrm{pic},5} = 8$ and $N_{\mathrm{pic}} = 12$.

If the MB impairment ratio in every layer is known, then the average MB impairment ratio can be calculated by

$$\mathrm{MBIR} = \frac{1}{N_{\mathrm{pic}}} \sum_{i=1}^{n} N_{\mathrm{pic},i} (\mathrm{MBIR})_i. \tag{13}$$

The problem now is how to find the number of self-induced impairments or vector $\bar{W}$ if the set of packet loss probabilities

$$\bar{\varepsilon} = \{\varepsilon_i\} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5) \tag{14}$$

is given. As indicated, the direct and indirect impairments are sometimes overlapped. To find the self-induced impairments vector $\bar{W}$, we first determine the direct impairments

$$\bar{\Omega} = \{\Omega_i\} = (\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5). \tag{15}$$

Given layer i and the packet loss probability ε_i, let $\varepsilon_j = 0$ if $i \neq j$, then the average number of impaired MBs in the layer at a GOP are the direct impairments.

A simple mathematical analysis can give a general and inaccurate solution to statistic problems. However, experimental investigation will provide a more accurate solution for specific problems and parameters. To determine the values of $\{\Omega_i\}$, we have experimentally investigated the programs BBC sequence 3, Cactus and Comb, Flowers, Mobile and Calendar, Susie, and Table Tennis. The experimental block diagram is shown in Fig. 5.

To make the investigation simple, the following conditions and simplifications are given:

1. Packets are randomly lost, with loss probability ε_i for the ith layer.
2. When a packet is lost, the lost data are replaced by a random binary sequence, not containing any *Psc* (start-code-prefix). This is to ensure a correct synchronization of the MPEG-2 streams.
3. Header data are not subject to packet losses in order to avoid frame skips and decoder failures.
4. All the packets in different layers have the same length of 512 message bits.
5. The macroblock PSNR is computed between the two reconstructed pictures: one with packet loss and one without packet loss as the original, as we are interested in the effect of packet-lossy transmission on the video quality.
6. We define that if MB PSNR $\leqslant$ 30 dB, the MB is an impairment, otherwise, it is not.
7. As commonly used, the picture size is 704×576, the color representing system is 4:2:0, the MB size is 16×16.

The experimental results are shown in Fig. 6. It can be observed that the direct impairments at different layers are linear functions of the packet loss probability if $\varepsilon \leqslant 0.1\%$ (the figure just shows a part, from 0% to 0.1%), that is,

$$\bar{\Omega} = \{\Omega_i\} = \{\eta_i \varepsilon_i\}, \quad i = 1, 2, \ldots, 5. \tag{16}$$

From the figure we have

$$\bar{\eta} = \{\eta_i\} = (29000, 13000, 13000, 13000, 39000) \tag{17}$$

for the given picture rate, picture size and MB size.

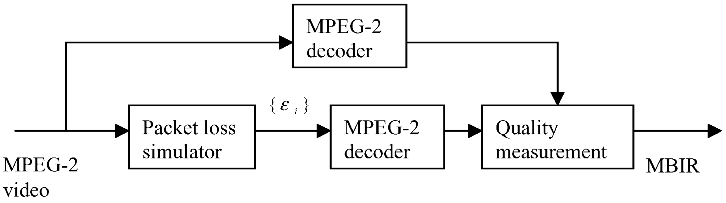


Fig. 5. Experimental block diagram for packet loss and video quality.

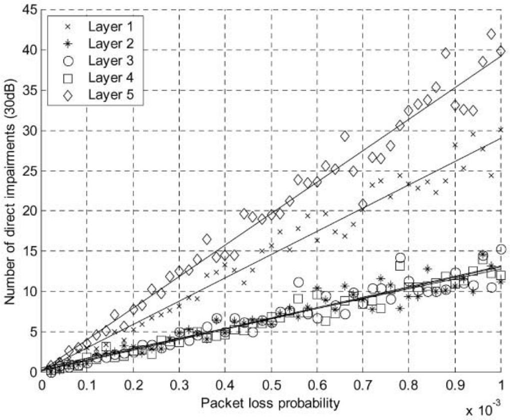


Fig. 6. Average number of direct impairments Ω_i in a GOP versus the packet loss probability ($N_{\mathrm{pic}} = 12$, $N_{\mathrm{pic},1} = \cdots = N_{\mathrm{pic},4} = 1$, $N_{\mathrm{pic},5} = 8$ and $N_{\mathrm{mk}} = 1620$).

Eqs. (16) and (17) show, for example, that if the I-picture loss probability is $\varepsilon = 0.1\%$, then in average there are 29 impaired MBs among the 1584 MBs on an I-picture. It is obvious that the $(\mathrm{MBIR})_1$ is much greater than the ε_1, because intra-coded DCT blocks have their DC coefficients coded differentially with respect to the previous DCT block of the same Y/Cb/Cr type in a slice [1]. When an I-packet is lost, the impairments occur not only on the corresponding MBs, but also on the following MBs in the slice.

In the first layer, all the impaired MBs are direct impairments. Therefore, the number of self-induced impairments is

$$W_1 = \Omega_1 = \eta_1 \varepsilon_1. \tag{18}$$

The impairments will result in indirect impairments in the upper layers, as shown in the propagation matrix. In the second layer the direct impairments can be estimated by $\Omega_2 = \eta_2 \varepsilon_2$ and the indirect impairments by $W'_2 = W_1 E_{12}$. Among Ω_2 and W'_2, there are overlapped microblocks. The probability of an overlapped impairment on a specific MB is about $(\Omega_2 W'_2)/(N_{\mathrm{mac}} N_{\mathrm{pic},2})^2$. Thus, the number of self-induced impaired MBs in the second layer equals Ω_2 minus the number of overlapped impairments, that is

$$W_2 = \eta_2 \varepsilon_2 \left(1 - \frac{W_1 E_{12}}{N_{\mathrm{mac}} N_{\mathrm{pic},2}} \right). \tag{19}$$

If we reasonably define $W_0 = 0$, the self-induced impairments can recursively be calculated by

$$W_i = \Omega_i \left(1 - \frac{1}{N_{\mathrm{mk}} N_{\mathrm{pic},i}} \sum_{j=1}^{i-1} W_j E_{ji} \right) \tag{20}$$

for $i = 2, 3, \ldots, n$ [12]. Consequently, the MBIR can be estimated by (11)–(13).

For example, given an MPEG-2 program with average data rate $D = 8$ Mbits/s, and the packet loss probabilities $\varepsilon_1 = 0.0005$, $\varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 0.005$ and $\varepsilon_5 = 0.01$, then we can find that

- direct impairments $\bar{\Omega} = (14.5, 65, 65, 65, 39)$ by Eq. (16),
- self-induced impairments in layers $\bar{W} = (14.5, 64.14, 60.11, 55.83, 34.37)$ by Eq. (18),
- total impairments in layers $\bar{\lambda} = (14.5, 85.5, 181.9, 284.4, 1573.5)$ by Eq. (11).
- MBIR in layers $\{(\mathrm{MBIR})_i\} = (0.0090, 0.0528, 0.1123, 0.1756, 0.1214)$ by Eq. (12),
- The average impairment ratio $\mathrm{MBIR} = 0.1101$ by Eq. (13).

The experimental results for the given packet loss probabilities are

- MBIR in layers $\{(\mathrm{MBIR})_i\} = (0.0090, 0.0637, 0.1167, 0.1634, 0.1373)$,
- the average impairment ratio $\mathrm{MBIR} = 0.1209$.

Obviously, the calculated results are very close to the experimental results.

5. Conclusion

In this paper we indicate that layered video signals can be modeled as a vector and a matrix. The vector represents the normalized data rates at the different layers, and the matrix represents the impairment propagation factors between the layers. Since motion vector for each macroblock is added to form a predictive picture, a lost packet will damage at least one macroblock (usually tens).

Therefore, the macroblock impairment ratio is suggested to be a measure of the video transmission quality instead of the most commonly cited metric PSNR. PSNR is useful in some applications, such as image coding, where the encoding process introduces degradation almost everywhere in the image, but the packet lossy channel introduces impairments in some areas of the reconstructed pictures. In this paper we also show that the average number of impaired macroblocks at the layers can be estimated by the product of the self-induced impairment vector and the impairment propagation matrix, where the values of the self-induced impairment vector are a function of the packet loss probability. Although the investigation in the paper is focused on MPEG-2, the principles and the methods are available for any layered video signals. They are useful for the development of efficient schemes and protocols for video signal layering and transmission over lossy networks or channels.

While the paper presents a mathematical model for the estimation of picture quality for unequal packet loss protection or packet priority dropping systems, it leaves many questions unanswered, for example, the values of elements in the self-induced impairment vector are experimentally obtained for a given packet length. The element values are also a function of data rate, packet length, average slice size, etc. A formula for the calculation of the values should be derived. In the paper we assume that the packets are randomly lost. Packet losses are generally correlated, which results in error bursts at the macroblock level. Our experiments have shown that random packet losses result in a larger macroblock impairment ratio than correlated packet losses do for the same loss probability, without consideration of the use of error concealment techniques. Anyhow, this problem should further be investigated together with error concealment and quality measure of reconstructed pictures. These issues are currently under investigation.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful and constructive comments.

References

[1] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, Priority encoding transmission, IEEE Trans. Information Theory (special issue devoted to coding theory) 42 (6) (November 1996) 1737–1744.

[2] ANSI T1.801.03-1996, Digital transport of one-way video signals – parameters for objective performance assessment, American National Standards Institute, 1996.

[3] S. Deering, Internet multicast routing: state of the art and open research issues, in: Multimedia Integrated Conferencing for Europé (MICE) Seminar at the Swedish Institute of Computer Science, Stockholm, October 1993.

[4] B. Girod, N. Farber, Feedback-based error control for mobile video transmission, Proc. IEEE 87 (10) (October 1999) 1707–1723.

[5] J. Hagenauer, T. Stockhammer, Channel coding and transmission aspects for wireless multimedia, Proc. IEEE 87 (10) (October 1999) 1764–1777.

[6] D. Hoffman, M. Speer, Hierarchical video distribution over internet style networks, in: Proceedings of IEEE International Conference on Image Processings, Lausanne, Switzerland, September 1996, pp. 5–8.

[7] ISO/IEC 13818-2, International standard ISO/IEC 13818-2 Information technology – Generic coding of moving pictures and associated audio information: Video, 1996.

[8] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[9] S. McCanne, M. Vetterli, Join source/channel coding for multicast packet video, in: Proceedings of IEEE International Conference on Image Processing, Washington DC, October 1995, pp. 25–28.

[10] O'Leary, D. Priestly, N. McSparron, Mobile broadcasting of DVB-T signals, IEEE Trans. Broadcasting 44 (3) (September 1998) 346–352.

[11] Tektronix FTP site, ftp://ftp.tek.com/tv/test/streams/Element/MPEG-Video/625/, 1999-07-31.

[12] T. Zhang, Y. Xu, Unequal packet loss protection for layered video transmission, IEEE Trans. Broadcasting 45 (June 1999) 243–252.

Paper II

Objective End-to-End QoS Gain from Packet Prioritization and Layering in MPEG-2 Streaming

Daniel Forsgren, Ulf Jennehag, and Patrik Österberg

Published in
Proceedings of the 12th International Packetvideo Workshop (PV 2002),
Pittsburgh PA, USA, April 2002.

Objective End-to-End QoS Gain from Packet Prioritization and Layering in MPEG-2 Streaming

Daniel Forsgren, Ulf Jennehag, Patrik Österberg

Mid-Sweden University
SE-851 70 Sundsvall
Sweden
{daniel.forsgren, ulf.jennehag, patrik.osterberg}@mh.se

Abstract

Layered video coding as well as prioritized packet scheduling are two well-known methods that may improve the quality of service level in real-time applications with high bandwidth requirements, and are used over packet switched networks. However, it is often difficult to get an idea of, and to quantify, the actual gains that may be achievable, especially from an end-to-end perspective.

In this paper, we present some experimental results from using temporally layered MPEG-2 video combined with basic per-layer IP packet prioritization. The goal has been to find out if a basic scheme is useful at all in combination with this particular source coding method, and if so, how much the objective video quality can be increased during bandwidth-constrained periods. The quality is measured in terms of PSNR and the results are compared to the case of equal packet priority. Also, different packet sizes as well as packet queuing disciplines are used.

We conclude that using even a relatively simple temporal layering strategy in combination with packet prioritization can quite significantly improve the end-to-end quality of MPEG-2 video, especially in moderately bandwidth constrained situations. Furthermore, packet size and queuing discipline is found to have a impact.

Keywords: MPEG-2, layered video, quality of service, traffic shaping

---

1 Introduction

Real-time video transmission over packet switched networks, such as the IP-based Internet, still remains a challenge after years of research. Realistic transmission of high-quality video streams generally requires high compression ratios, which lead to an increased sensitivity to transmission errors (e.g. packet loss) and make this case especially challenging.

Successful approaches to eliminate or reduce network degradation of real-time media streams include addressing the problem at, or below, the network (IP) level by using physical links with quality of service (QoS) support, enhanced QoS-aware routing protocols [1], support for integrated services/differentiated services etc. In practice, many of these approaches require upgrades of network nodes of a kind that network operators typically are reluctant to introduce, for reasons of complexity or economy. Today, most IP networks remain best effort, although support for quality of service functionality will most likely grow in proportion with the interest for using high-quality media applications over the networks.

The lack of network support for QoS could be partially dealt with using lesser means, by making the best effort of the network even better for some applications or traffic types. This type of solution tries to reduce the impact of any congestion problems closer to network leafs, without relying (at least not exclusively) on QoS functionality in the network as a whole to take care of traffic prioritization. Traffic shaping [2] is often implemented in individual hosts for shaping their outgoing traffic, but it can also be performed on interfaces further into the network.

For example, a router that forwards traffic from a subnet in which many real-time video streams are known to originate, ending up in the aggregated traffic flow on its interface(s), could retain instantaneous information about the currently available bandwidth in each direction. Based on this information (and perhaps even information about throughput along a whole network path) the router could adjust the packet prioritization behavior on each interface according to prevailing local and/or path conditions. The more routers along a communication path that would use this strategy, the better the bandwidth utilization would be. With such a scheme, if a router approaches its current bandwidth limit on a certain interface, it can for example, elect to drop more packets belonging to non-real time applications.

However, in a router where there are lots of aggregated high-bandwidth video data flows, there is a risk that most of the data would end up belonging to the prioritized group, and the prioritization would become meaningless. For the prioritization to be successful in these situations, the router should be able to make more fine-grained prioritization decisions than what is typically feasible. Preferably it should be able to distinguish between, and make prioritizations of, different types of data within a video stream (e.g. several traffic classes per application). This level of granularity is not feasible to implement throughout a large network, but in routers close to leaf networks where the aggregated traffic level is moderate, it can be.

One way of achieving this granularity is to employ a layered source coding method. This would enable nearby network nodes to prioritize certain data inside each video stream. Using layer (i.e. packet) priority information, as well as information about local and perhaps even remote bandwidth availability, routers can queue and drop packets in a “controlled” manner before congestion occurs.

In this paper, we present the results of experiments made using temporally layered source coding in combination with an intermediate router node, which uses class-based queuing to give different priority to packets depending on which video layer their payload belongs to. The goal of the experiments was to see how much the overall end-to-end video quality, in terms of luminance peak signal to noise ratio (PSNR) [3], could be improved in moderately bandwidth-constrained situations using simple variants of such an approach.

2 Layered Video Coding and Transmission Method

Scalable and layered video transmission schemes have been widely studied before [4, 5]. However, their actual use in combination with different fine grained packet prioritization schemes and the improvement in perceived end-to-end video quality that can be achieved, has not received as much coverage. This chapter introduces the video coding format, layering method and transmission methods that we have used throughout our experiments.

The video-coding format used is the widely adopted MPEG-2 [6]. We have used multiple pre-encoded MPEG-2 streams, each with a constant bit rate (CBR) of

8 Mbps, consisting of the popular reference sequences “BBC3”, “Cactus and comb”, “Flower garden”, “Mobile and calendar”, “Susi”, and “Table tennis”, all of which are available from the Tektronix FTP site [7]. The pre-encoded streams are divided into video layers offline, while the transmission and reception of the layered video streams take place in real-time.

The temporal layering method we have used assigns the different picture types in a group of pictures (GOP) to a certain layer depending on the importance of the picture type. Even though the GOP structure is not mandatory in MPEG-2, the need of a repeating pattern is required in order to apply a temporal layer method. Therefore the GOP structure of twelve pictures per GOP, of which three are P pictures, is assumed. Figure 1 shows the GOP structure and how it has been mapped to the video layers. The importance of the layers is in numerical order where layer zero is most important.

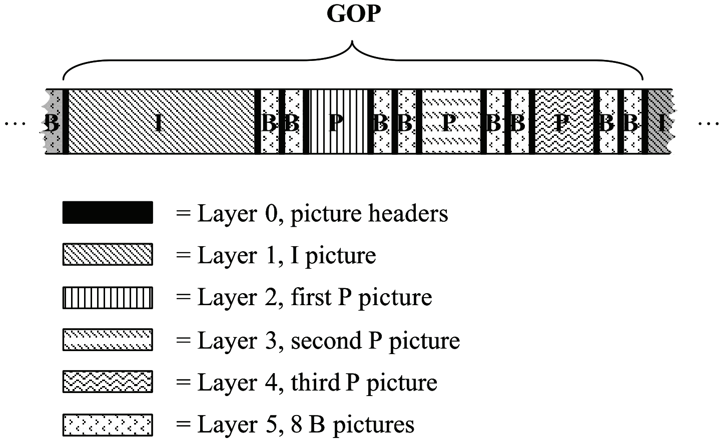


Figure 1: The GOP structure and its mapping to video layers

More accurate approaches exist for determining the packet priority. It can be calculated from the macroblock (MB) encoding type, the motion vectors (MV) of each MB, total packet size and picture-level header occurrences [8]. The MVs are relevant since their length and direction affect the error propagation [9]. Another parameter that may be relevant to consider is the dependence count, the number of times a MB is referenced.

Our experiments have been performed in a setup, laid out as shown in figure 2. The sending application, running in the send host, aims at transmitting each of the different priority classes with CBR. That is, even though in the video stream, there might be a burst of consecutive packets belonging to the same priority class, these packets are distributed evenly among the other packets when transmitted. We consider this per-GOP packet interleaving a good measure, since it does not incur any significant additional delay, and reduces the need for buffering bursts in network queues when there are few parallel video streams. Due to the chosen re-distribution of the packets, a real time streaming video receiver will experience a short initial delay, which corresponds to the time it takes to receive all the packets belonging to the first GOP.

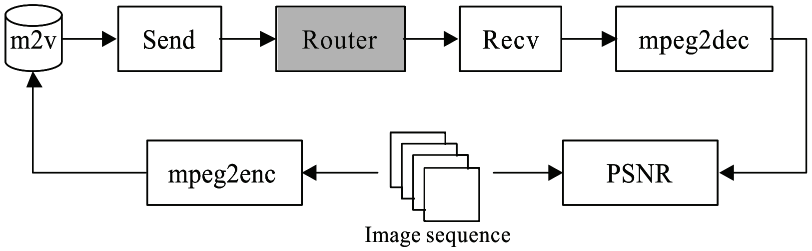


Figure 2: Experimental setup

For the receiver to be able to process packets arriving in this non-deterministic order, the sender adds a sequence number (SN) at the beginning of each packet, thus adding 4 bytes of overhead. The sender transmits video data using UDP/IP. Therefore, besides the payload of video data of each packet, the packets will consist of 8 bytes of UDP header and another 20 bytes of IP header. The layout of a video packet can be seen in figure 3.

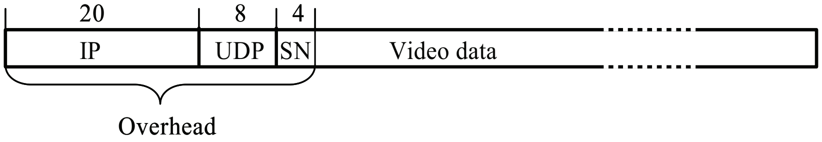


Figure 3: Video packet layout

When all the video data packets that make up a clip have been transmitted, the sender transmits an end of transmission (EOT) packet, which is always delivered to the receiver. This is only for making the experiments, where we send thousands of clips in rapid succession, easier to track, and would not necessarily be needed in real world applications.

As the receiver receives a packet, it extracts the SN and copies the video data in the packet payload to the corresponding position of an assigned memory area. When an EOT packet is received, the receiver copies the data from the memory area to a file. The PSNR of the video in the file is later calculated off-line using the original image sequence as a reference. The decoder used in the experiments is the MSSG MPEG-2 video codec [10]. The decoder has been slightly modified to be able to decode erroneous MVs and illegal variable length codes (VLC), functionality to handle loss of picture headers has also been included.

3 Network Setup and Parameters

As described in section 2, a sender host has acted as a video server, and transmitted video clips separated into temporal layers, encapsulated in UDP/IP packets. On their way to the recipient, the packets have been routed through an intermediate router node, in which the outgoing interface has been configured for class-based traffic control. The network interfaces involved in the experiments were all of fast Ethernet type, with a maximum transmittable unit (MTU) of 1500 bytes. The router node has been using Linux 2.4 kernel traffic filtering and control functionality to map incoming packets to the proper traffic class on the outgoing network interface, and queued accordingly. The mapping has been based on IP header information, e.g. the type of service (TOS) value.

For the division of outgoing traffic into traffic classes, the router has been using hierarchical token bucket (HTB) class based queuing, which is a more efficient and clean alternative to the standard class based queuing (CBQ) [11] that is a component of the Linux 2.4 kernel family [12]. The packet queuing disciplines we have chosen to use and compare, in both the prioritized and the non-prioritized cases, are simple first in first out (FIFO) and stochastic fair queuing (SFQ). FIFO requires little introduction, SFQ was discussed in [13] and is basically a lightweight fair queuing discipline that uses collected packet statistics to shape the traffic.

FIFO and SFQ are two of the most popular disciplines available, and they are also suitable for use in our proposed scenario (in leaf routers with limited aggregated bandwidth). Furthermore, neither one of these disciplines are controlled by a large number of parameters, which makes them good candidates for objective comparison. In contrast, many of the more complex queuing disciplines, such as random early detection (RED) (which is primarily designed for use in backbone routers) have a large number of parameters, which are typically chosen and tuned using approximate rules-of-thumb to obtain the desired behavior. In our experiments, however, we have kept all queuing parameters fixed for all experiments.

In the network, the variables in the experiments have been the total available bandwidth, whether packets are prioritized or not, and which queuing strategy is used. The network had no notion of packet size or statistics of the particular video clip currently being transmitted.

In the non-prioritized cases, we have simply used a single traffic class (set for either FIFO or SFQ queuing), into which all data packets have been filtered, irrespective of which layer the data happens to belong to. The total available bandwidth in the single class has ranged between 5000 and 8800 kbps in the experiments, with less spacing between data points in the more interesting areas of the bandwidth interval. Since we have used a video compression scheme that is very sensitive to data loss, we have judged this area to reside in the upper quarter of the bandwidth interval.

In the prioritized cases, we have used a prioritization scheme where packets are filtered into a layer-specific traffic class, depending on the type of data the packet is carrying as indicated by the IP header. The total bandwidth range available to all the classes was the same as in the non-prioritized case. To achieve an efficient, yet simple and practically feasible, prioritization between the traffic classes, we let the classes carrying information from high-priority video layers get a guarantee of having a greater percentage of their expected (average) bandwidth available when needed. Our chosen values for guaranteed bandwidth are shown in table 1.

Layer	Content	Average bw	Guaranteed bw
0	Header data	280kbps	100%
1	I-frames	2100kbps	90%
2	P-frames 1	900kbps	70%
3	P-frames 2	900kbps	50%
4	P-frames 3	900kbps	30%
5	B-frames	3000kbps	20%

Table 1: Layer prioritization

Each class has been allowed to use more bandwidth when available. However, no bandwidth has been considered excessive before all classes have obtained required bandwidth up to their respective guaranteed levels. No class is allowed to use bandwidth exceeding a certain peak level, which has equaled the total available bandwidth in each experiment. In order to get a consistent queuing behavior, the sum of the guaranteed bandwidths for all classes has been ensured to stay well below the minimum available total bandwidth (i.e. the bandwidth of the main parent class) in all experiments.

The average bandwidth figures for the video layers have been calculated over all the video clips and packet sizes used. We have found these figures, and especially the relation between them, to be representative enough for all the scenarios, thus we have kept them constant for all experiments.

The output was in the form of a number of received video clips, which were finally decoded and used to calculate the average PSNR impact of the network conditions. Each experiment has been performed a large number of times and with all of the six video clips [7], in order to improve the accuracy of the results. All experiments have also been repeated for each of the three fixed packet sizes (550, 1000 and 1450 bytes payload).

4 Results

Figure 5 shows an overview of the PSNR performance under each of the four cases. The payload per packet is in this case fixed to 550 bytes. It is evident that the layer prioritization creates an overall lift of the PSNR curves when compared to the non-prioritized case with the same packet queuing strategy. Figure 5 also shows that using SFQ improves the video quality when compared to FIFO, especially in the non-prioritized case.

Figure 6 shows the corresponding results with the only difference being that a payload of 1000 bytes has been used in each video packet. It can be clearly seen that increasing the packet size has a positive impact on the PSNR performance in all bandwidth-constrained situations, and the prioritized case is still superior. The increase in PSNR from the use SFQ instead of FIFO is approximately doubled.

Finally, in figure 7 the corresponding plots for the largest packet size used, 1450 bytes payload, can be found. As expected, the PSNR performance is best in this case. SFQ has an even greater influence than in the smaller packet cases, probably because it was configured for an MTU of 1500 bytes.

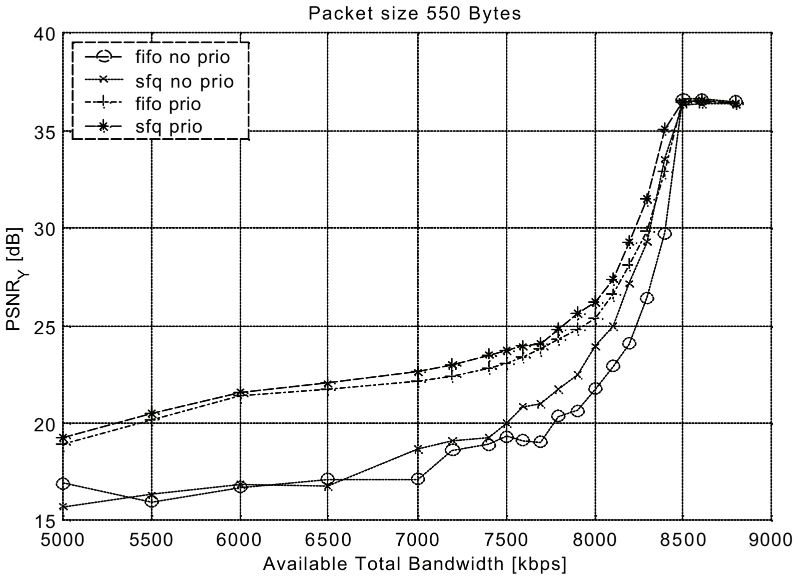


Figure 5: Performance of each scheme with 550 bytes payload

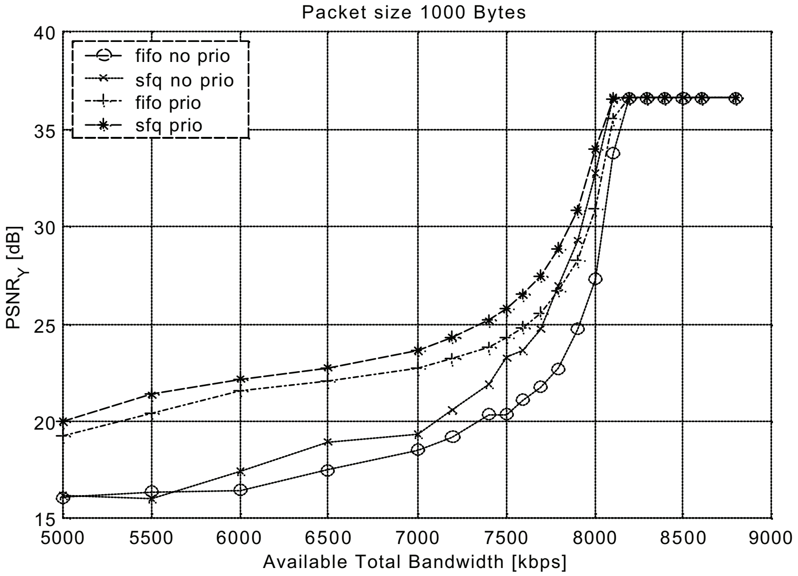


Figure 6: Performance of each scheme with 1000 bytes payload

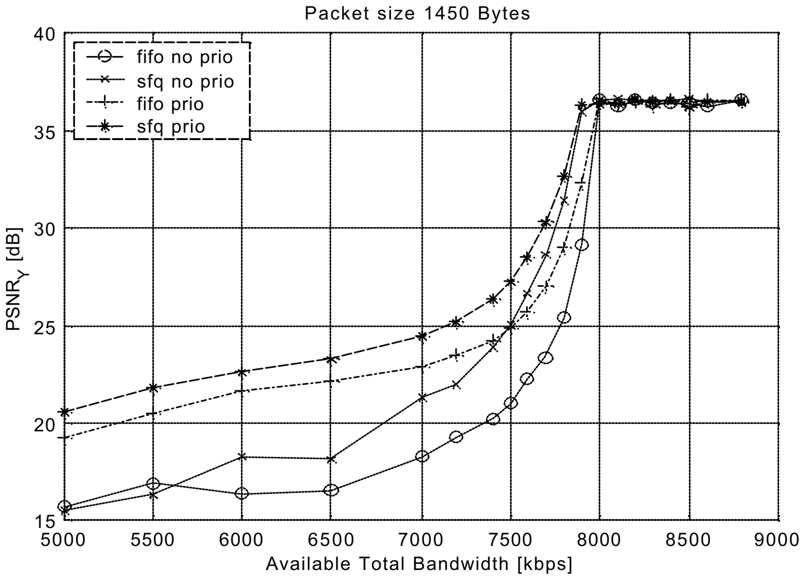


Figure 7: Performance of each scheme with 1450 bytes payload

5 Conclusions and Future Work

The results of these experiments show an increase in objective video quality, in terms of PSNR, when packet prioritization is applied. At lower available bit rates, 1 Mbps or more below the bit rate of the sender, this increase is about 3-5 dB. At bit rates closer to the bit rate of the sender, the increase is about 2-3 dB.

We can also see that the queuing strategy affects the end-to-end video quality; SFQ outperforms FIFO. At bit rates closer than 500 kbps to the bit rater from the sender, this influence is greater than that of the prioritization.

Furthermore, it is evident that packet size also has an impact on the objective video quality, and should be chosen as large as possible. An obvious reason for this is that a smaller packet size leads to extra overhead, but the difference is too big to be solely explained by the extra overhead. Another reason is likely the fact that with an error sensitive source coding schemes like MPEG-2, it is due to error propagation better to lose large chunks of data at a lower rate than small chunks at a higher rate.

Future work could include to make similar experiments using other source coding schemes (e.g. MPEG-4) together with more advanced and fine-grained layering methods (e.g. spatial scalability coupled with temporal scalability), and a greater variety of network settings, in order to further increase the overall end-to-end PSNR efficiency. Optimally, the PSNR curve should get a more linear behavior over the entire bandwidth spectrum. This is dependent on all tiers in the layered video transmission system, but the layered source-coding scheme is probably the most important.

It would also be interesting to make experiments where multiple layered video streams are sent over the same data prioritizing link, and to add competing background traffic to the aggregated traffic flow, to test the scalability of the approach in terms of PSNR.

References

[1] Chen, S. and Nahrstedt, K., "An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions", IEEE Network Magazine, Vol. 12, No. 6, November-December 1998

[2] Bechler, M., Ritter, H., Schafer, G. and Schiller, J. 'Traffic shaping in end systems attached to QoS-supporting networks" Proceedings of the Sixth IEEE Symposium on Computers and Communications, 2001

[3] ANSI T1.801.03-1996, "Digital transport of one-way video signals-parameters for objective performance assessment", American National Standards Institute, 1996

[4] Zhang, T. and Xu, Y., "Unequal packet loss protection for layered video transmission", IEEE Transactions on Broadcasting, June 1999

[4] Amir, E., McCanne, S., and Vetterli, M., "A layered DCT coder for Internet video", Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland, September 1996

[6] ISO/IEC 13818-2, "Information technology - Generic coding of moving pictures and associated audio information: Video", 1996

[7] Tektronix FTP site, ftp://ftp.tek.com/tv/test/streams/Element/MPEG-Video/625/, July 31,1999

[8] Jitae, S., JongWon, K. and Kuo, C.-C. J., "Relative priority based QoS interaction between video applications and differentiated service networks", Proceedings of International Conference on Image Processing, Vancouver, September 2000

[9] de los Reyes G., Reibman, A., and Chang, S., "A corruption model for motion compensated vide subject to bit errors" Proceedings of Packet Video Workshop, New York City, May 1999

[10] MPEG Software Simulation Group (MSSG), "MPEG-2 Encoder / Decoder, Version 1.2", ftp://ftp.mpeg.org/pub/mpeg/mssg/, July 19, 1996

[11] Devera, M. and Cohen, D., "HTB Linux queuing discipline manual", http://luxik.cdi.cz/~devik/qos/htb/, December 19, 2001

[12] Hubert, B. et al, "Linux Advanced Routing & Traffic Control HOWTO", http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html, December 6, 2001

[13] McKenney, P., "Stochastic Fairness Queuing", Proceedings of INFOCOM, San Fransisco, June 1990

Paper III

Increasing Bandwidth Utilization in Next Generation IPTV Networks

Ulf Jennehag and Tingting Zhang

Published in
Proceedings of the 2004 International Conference on Image Processing (ICIP 2004),
Singapore, 2004

2004 International Conference on Image Processing (ICIP)

INCREASING BANDWIDTH UTILIZATION IN NEXT GENERATION IPTV NETWORKS

Ulf Jennehag and Tingting Zhang

Teleinformatics Research Group (ITM/IKS), Mid-Sweden University, SE-851 70 Sundsvall, Sweden
Ulf.Jennehag@mh.se Tingting.Zhang@mh.se

ABSTRACT

In this paper we present a novel idea regarding transmission of next generation IPTV content. Today IPTV systems utilize a GOP structure to provide stream synchronization points for the clients, these are used when a channel switch occur. Since channel switches made by TV audiences are quite rare it is redundant to send synchronization opportunities in a GOP manner. The proposed design, synchronization frames for channel switching (SFCS), only requests synchronization frames when needed. We present a network traffic analysis model and compare it with simulations. SFCS increases bandwidth utilization compared to traditional GOP system under the presented transmission environment.

1. INTRODUCTION

In recent years the development of high-speed metropolitan networks has exploded, more and more households have access to such a network. The increasing number of actors (i.e. ISP and net owners) providing this service lowers the cost, this pressure these actors to introduce new services to increase revenue. Examples of such services are IP-telephony, IPTV and VoD.

In the case of IPTV there exists a need for an efficient way of transporting digital video over an Internet like network in a broadcast fashion. IP-Multicast is a proven way to efficiently transmit multimedia content to a selected set of receivers. The issue of transmitting multimedia via IP-Multicast and related questions has been under research for several years [1]. An IPTV stream (channel) consists of video, audio (could be several streams), and additional information. However, the video part of the stream is dominating.

For an IPTV transmission system to be economically feasible in an Internet-like environment a reasonably large number of clients must be connected; in addition to this there must be a sufficient number of channels available.

A widely spread solution to this problem is to use broadcast material in a Multicast environment using an existing video coding standard [2] over a well established transport system [3]. Such systems are in use all over the world today.

In such a system it is necessary to provide a system for a new client to synchronize to a stream. In most cases this is done by utilizing a video coding scheme with a group of pictures (GOP) structure. The GOP structure provides a reoccurring event of resynchronization points called Intra coded frames (I-frames). I-frames can be decoded without knowledge about the previous frames. To increase coding efficiency modern video coding schemes also use motion compensated predictive coded frames (P-frames). The additional redundancy the I-frame is carrying is only needed when the client have no information about the prior frame at all, i.e. the client is starting the decoding process of the selected stream. This occurs typically then the client is changing stream (i.e. channel). A study [4] of the channel changing behavior of a typical TV audience reports that the average number of changes in one hour of TV watching is quite small, about five times. Thus, transmitting I-frames to all users because one client is changing channel is a waste of bandwidth.

In this paper we propose a novel idea for the next generation IPTV networks. We propose a design that provides one multicast stream consisting of P-frames only, accompanied by a stream of synchronization frames. When a synchronization frame is needed it is transmitted on a separate multicast address and received only by the users who actually request it. In this paper we only focus on the video part of the IPTV data. All data received by a client is considered to be rendered immediately.

The paper is organized like follows; in section 2 and 3 we propose a novel idea regarding the transmission of digital video content for a broadcast environment like an IPTV service, in section 4 we present simulation results and finally in section 5 we present conclusions and future work.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

2. SYNCHRONIZATION FRAME FOR CHANNEL SWITCHING

Our proposed design, Synchronization Frame for Channel Switching (SFCS) is based on the idea that it is redundant to send intra coded frames at a fixed frequency to provide stream resynchronization points. With SFCS a stream consisting of resynchronization points (S-frames) is separated from the main stream (P-frames only). Clients (receivers) who want to decode the video stream (in this case a TV-channel) must receive the both the main stream and the synchronization stream. However, the synchronization stream is only received at the actual synchronization point and is later left. The streams are spliced and sent to the decoder. The decoded frames at a synchronization point must produce the same result in order to not produce mismatch and drift. One technique capable of this is switching pictures (SP/SI-frames) [5]. Although the concept SP-frames have been researched before [6], and is part of a standard [7], it has been very little or no work done on the technique applied to an IPTV environment.

Since the next generation IPTV technology probably will be based on modern compression scheme where SP/SI-frames, or similar, are included. In such system it should be possible to implement SCFS with success.

In this paper we focus on two types of frame coding, synchronization frame (S-frame) and predictive frame (P-frame). An S-frame at time *t* produces the same decoded content as the P-frame at time *t*.

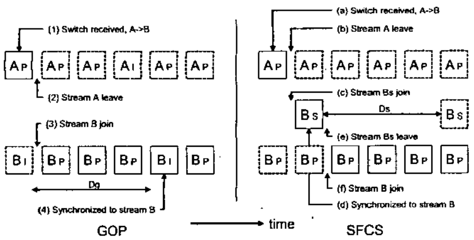


Figure 1: Channel switch with both GOP and SFCS

The following example (Figure 1) illustrates a client performing a typical switch from the present channel to another (i.e. A->B); in order to compare performance the same switch is also made in a GOP system.

In the GOP system the client requests a switch at (1). The client leaves the A stream (2) and joins the B stream (3). The decoder waits for the next I-frame in order to synchronize with the stream. The distance between these synchronization points (I-frames) equals Dg.

For the SCFS system the client requests the switch at (a), the A-stream is left at (b) and the synchronization stream (S-frames) for B (Bs stream) is joined (c). The client is now synchronized to steam B (d). The distance between two S-frames is Ds. The client leaves the Bs stream (e) and joins the main B stream (P-frames) (f).

The main feature of SFCS is that the link bandwidth for a specified channel is reduced due to the low number of synchronization frames needed compared to the amount sent when utilizing a GOP scheme.

Factors that govern the efficiency of SFCS are; number of receivers, number of selectable channels, channel popularity, frequency of synchronization frame offers, and the duration between channel switches.

3. NETWORK TRAFFIC

In this section, we compare the bandwidth requirements of SFCS and the traditional method (GOP). We consider a network as a set of links connected at nodes. A link can either be router-to-router link (RRL) or a router-to-client link (RCL).

The RRL carries the video streams between routers (or between the video server and the first router) and the RCL delivers the selected streams to a single client. Routers are considered to be Multicast enabled.

For the reader's convenience, we provide a list of all the variables used:

fps	number of frames per second
d	average duration in seconds between channel changes for a client
α	Zipf distribution parameter (0-1)
R_S	number of resynchronization points for SFCS during a second
R_G	number of resynchronization points for GOP during a second
P	average P-frame size in bits
S	average S-frame size in bits
r_{SP}	size ratio, S-frame size/P-frame size
M	number of available channels
N	number of receivers

3.1 Bandwidth requirement for the RCL

The average bandwidth requirement for the SFCS approach is

$$B_{SC} = \left(fps + \left(\frac{r_{SP}}{d} \right) \right) P \qquad (1)$$

And for a traditional GOP approach it is:

$$B_{GC} = (fps - R_G + R_G \cdot r_{SP})P \qquad (2)$$

3.2 Bandwidth requirement for the RRL

Let the probability of change to channel q is p_q. The process of channel change is considered to be Poisson distribution distributed. The possibility of one receiver changing to channel q between two resynchronization points is $\frac{p_q}{R_S \cdot d}$. The average number of receivers changing to channel q between two resynchronization points is $\frac{N \cdot p_q}{R_S \cdot d}$.

The possibility of there is no receiver changing to channel q between two resynchronization points is $e^{-\frac{N \cdot p_q}{R_I \cdot d}}$. The average number synchronization frames per second is $R_S \cdot \sum_{s=1}^{M}\left(1 - e^{-\frac{N \cdot p_q}{R_S \cdot d}}\right)$. So the average Bandwidth requirement of RRL is:

$$B_{SR} = \left(fps \cdot M + r_{SP} \cdot R_S \sum_{s=1}^{M}\left(1 - e^{-\frac{N \cdot p_q}{R_S \cdot d}}\right)\right)P \qquad (3)$$

In our simulation results (section 4) and in the network analysis calculations we use Zipf's distribution [8] with parameter 0 (α=0) and uniformly distributed channel popularity. The true channel popularity distribution is argued to be somewhere in between these.

In the case of channel popularity being uniformly distributed, we have

$$p_q = \frac{1}{M} \qquad (4)$$

The average bandwidth for the SFCS approach is

$$B_{SRU} = \left(fps \cdot M + r_{SP} \cdot R_S \cdot M \cdot \left(1 - e^{-\frac{N}{R_S \cdot d \cdot M}}\right)\right)P \qquad (5)$$

When channel popularity is Zipf distributed, we have

$$p_q = \frac{1}{q^{1-\alpha} \cdot C} \quad for\ all\ q = 1, \ldots, M \qquad (6)$$

where,

$$C = \sum_{j=1}^{M} \frac{1}{j^{1-\alpha}} \quad for\ all\ j = 1, \ldots, M \qquad (7)$$

From equation (6), (7) and (3), we get the following traffic bandwidth expression:

$$B_{SRZ} = \left(fps \cdot M + r_{SP} \cdot R_S \sum_{q=1}^{M}\left(1 - e^{-\frac{N}{R_q \cdot d \cdot q \cdot C}}\right)\right)P \qquad (8)$$

For the traditional GOP approach, we get the following bandwidth expression:

$$B_{GR} = M(fps - R_G + R_G \cdot r_{SP})P \qquad (9)$$

4. SIMULALTION RESULT

A simulation environment was created to compare with the theoretical results. The environment consists of a program simulating traffic flow in a predefined fictive multicast broadband network. The used network layout is described in Figure 2. Each client is connected to a router and is tuned to one channel (video stream) at a time. The popularity of the channels is either uniform or Zipf distributed with parameter 0. Group management messages (IGMP) and related traffic are considered to propagate and take effect immediately. In all the simulation and theoretical results the value of 3 is used for r_{SP}.

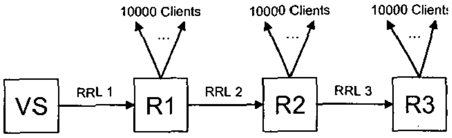


Figure 2: Simulated network

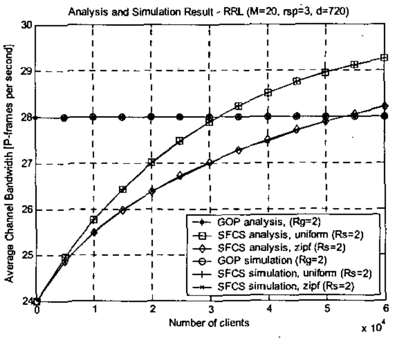


Figure 3: Analysis and simulation results.

The simulation program simulates the traffic load in the RRL (i.e. RRL 1-3) between the server and the connected routers. Figure 3 show the correlation between our theoretical model and the simulation results with only one RRL. Since the results are nearly identical, we only present theoretical results in our following figures. Figure 4 and 5 show that the performance for the proposed scheme is better than traditional GOP, even for a relatively low number of selectable channels. If the number of selectable channels is low or if the number of users is high, the number of resynchronization requests per channel and per second increases and approaches Rs. In this case SFCS can consume more bandwidth than GOP.

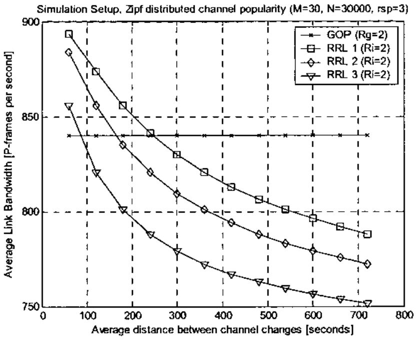


Figure 4: P-frames per second for RRL in the simulation setup (Figure 2) with varying average channel change duration, the number of selectable channels is 30.

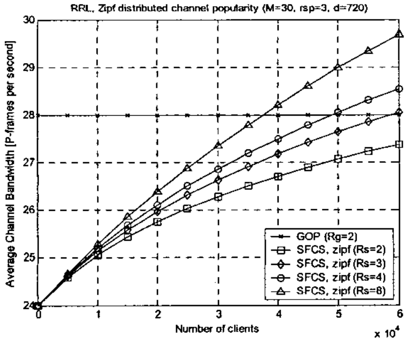


Figure 5: RLL client sweep with different Rs settings.

By changing the number of available resynchronization points per second (i.e. Rs), the time between the actual switch and until the receiver is synchronized to the stream (Ds in Figure 1). However, as mentioned before, and as illustrated in Figure 5 this influences the performance of the system.

5. CONCLUSIONS

In this paper we presented a novel idea using synchronization frames for channel switching (SFCS). SFCS is proved to significantly increase bandwidth utilization, in terms of average channel bandwidth, on RRL's even for a large number of users and a relatively small set of selectable channels when compared to a traditional GOP based IPTV system. SFCS can also decrease the waiting time between channel switches by increasing the frequency of S-frames per second. Since splicing of the streams can be conducted in a pre-decoder manner the actual implementation of the decoder is not an issue, as long as the requirements for SFCS are met.

Efficiency of SFCS is dictated by; number of receivers, number of selectable channels, channel popularity, frequency of synchronization frame offers, and the duration between channel switches.

5.1 Future Work

The following topics are of interest for future work:

- Error concealment, SFCS provides means for stream resynchronization when packet loss is detected.
- SFCS is inefficient for popular channels or when switching occurs to often, a hybrid scheme with GOP and SFCS is of interest.

6. REFERENCES

[1] Wu D., Hou Y., Zhu W., Zhang Y-Q., and Peha J.,"Streaming Video over the Internet: Approachs and Directions", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, March 2001

[2] ISO/IEC 13818-2, *Information technology - Generic coding of moving pictures and associated audio information: Video*, 1996

[3] ISO/IEC 13818-1, *Information technology - Generic coding of moving pictures and associated audio information: Systems*, 1996

[4] Brennan M. and Syn M.,"Television Viewing Behavior During Commercial Breaks", 2001 Australian & New Zealand Marketing Academy Conference (Anzmac 2001), Dec 2001

[5] Karczewicz M., Kurceren R.,"The SP- and SI-frames design for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, Issue. 7, July 2003

[6] Farber N., Steinbach E., Girod B.,"Robust H.263 Compatible Transmission for Mobile Video Server Access", First International Workshop on Wireless Image/Video Communications 1996, 4-5 Sep. 1996

[7] ITU-T Rec. H.264, *Advanced Video Coding for Generic Audiovisual Services*, 2003

[8] G. Zipf, *Human Behavior and the Principle of Least Effort* ,Addison-Wesley, Reading, MA:, 1994

Paper IV

Improving Transmission Efficiency in H.264 Based IPTV Systems

Ulf Jennehag, Tingting Zhang, and Stefan Pettersson

Published in
IEEE Transactions on Broadcasting, Volume 53, Issue 1, Pages 69-78,
March 2007.

Improving Transmission Efficiency in H.264 Based IPTV Systems

Ulf Jennehag, Tingting Zhang, *Student Member, IEEE*, and Stefan Pettersson, *Member, IEEE*

***Abstract*—In this paper, a novel proposal regarding the transmission of the next generation of live-TV content in an IPTV environment is presented. Today live-TV IP transmission systems utilizes a Group Of Pictures (GOP) structure in order to provide stream synchronization points for the clients. Synchronization points are used when a TV-channel switch occurs. The number of channel switches made by TV audiences have been shown to be rather rare. It is therefore redundant to send synchronization events in a GOP manner. Our proposal, Synchronization Frames for Channel Switching (SFCS), only requests synchronization points when required. Due to channel popularity distributions and the number of connected users, the total number of synchronization requests for a popular channel can increase to a level that makes SFCS less effective than GOP. Therefore, we introduce an SFCS-GOP hybrid. A complete network traffic analysis model, verified by a simple simulation environment, is also presented and the results show that the SFCS-GOP hybrid significantly increases the bandwidth utilization compared to a traditional GOP system.**

***Index Terms*—Interactive TV, multicast channels, video coding.**

I. Introduction

OVER RECENT years there has been an increased interest for video transmission services over packet-switched networks. Broadband TV [1] or Internet Protocol Television (IPTV) [2], [3] are two names used to describe a collection of TV-services over IP broadband networks including, live-TV, Video on Demand (VoD) as well as other interactive services. Triple play is a widely used term for a bundle of services that often include IPTV, Internet access and IP-telephony. In recent years, the development of high-speed metropolitan networks has grown rapidly and as a consequence, more households have access to such networks. This notion is confirmed by the 2005 issue of the annual report “Bredband i Sverige” by Post och Telestyrelsen in Sweden [4]. In addition, Informa Telecoms & Media reports that the market for IPTV is expected to increase tenfold over the next five years [5]. This makes IPTV an interesting research topic for the future.

A typical live-TV stream (channel) consists of video, audio, and additional information such as channel information. However, the video part of the stream is dominates in terms of data rate. In this paper we propose a method for increasing the efficiency in transmitting live-TV content in an IPTV environment. Hence, the main focus of the paper is on the live-TV part of IPTV.

For an live-TV IP transmission system to be economically feasible, a reasonably large number of clients must be connected and there must be a sufficient number of channels available. A large number of available channels places significant demands on the underlying transport network as well as transmission protocols. A widely used solution involves broadcast material in a multicast enabled environment through an existing video coding standard, like MPEG-2 [6] packetized in a MPEG-2 Systems Transport Stream (MPEG-2 TS) [7]. A similar approach is discussed in [8]. IPTV systems are today in use all over the world. Two examples are FastWeb in Italy [9] and FastTV in Sweden [10]. There exist some solutions today offering H.264 in an IPTV environment using MPEG-2 TS over IP as transport protocol.

The need for synchronization to the stream occurs typically when the receiver of the stream (client) is changing video stream, i.e. TV-channel, or if the client loses synchronization for some other reason such as packet loss introduced by network congestion. A study by Brennan *et al.* [11] of the channel changing behavior of a typical TV audience reports that the average number of changes occurring in one hour is only about five times. Thus, transmitting intra coded frames to all receiving users is actually providing a somewhat inefficient solution when the actual requirements for stream synchronization are taken into account. Similar approaches have been presented before [12], however, the investigations lack analysis depth. This paper is based on earlier work by the authors [13].

A. Video Coding Frame Types

The GOP structure provides recurring resynchronization points called Intra coded frames (I-frames). I-frames can be decoded without any knowledge concerning the previous frames. To increase the coding efficiency, modern video coding schemes also use motion compensated predictive coded frames (P-frames) and, additionally, bidirectional predicted frames (B-frames) to further increase the coding efficiency.

B. Switching Frames

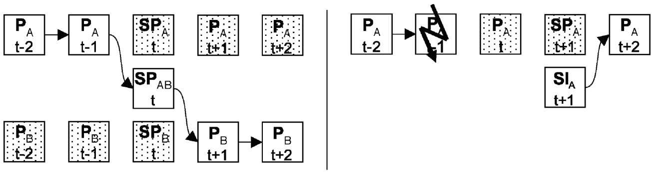
The concept of switching frames (SI/SP) was introduced by Karczewicz and Kurceren [14] and is based on the initial work by Färber *et al.* [15]. Switching frames can be used to switch between different encoding representations [16] of a sequence and allowing random access to a stream with only predictive coded frames. Fig. 1(a) shows how switching frames can be used to switch between two bit-streams. White frames are received by the decoder and grey frames are not. A decoder tuned to stream A can use a transition frame, i.e. $\mathrm{SP}_{\mathrm{AB}}$, to change to stream

---

Manuscript received June 29, 2006; revised October 13, 2006. This work was financed in part by the Regional Fund of the European Union, and under the County Administrative Board of Västernorrland (Y3041-207-05).

The authors are with the Department of Information Technology and Media, Mid Sweden University, S-851 70 Sundsvall, Sweden (e-mail: Ulf.Jennehag@miun.se; Tingting.Zhang@miun.se; Stefan.Pettersson@miun.se).

Digital Object Identifier 10.1109/TBC.2006.887167



(a) (b)

Fig. 1. (a) Using SP-frames to switch between bits-streams. (b) SI-frames in a random access situation.

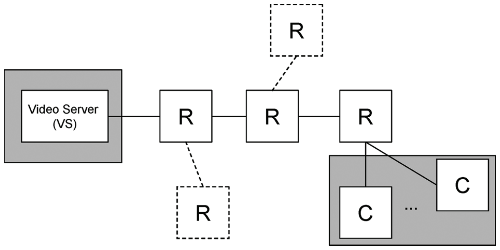


Fig. 2. A typical IPTV system, with video server, routers and clients.

B. When the $\mathrm{SP}_{\mathrm{AB}}$-frame at time t is decoded, it produces the same result as the SP_{B}-frame at time t. Hence, these two frames produce exactly the same result and can thus be used as a base for accurate prediction decoding P_{B} at $t+1$. Note that $\mathrm{SP}_{\mathrm{AB}}$ not necessarily equals $\mathrm{SP}_{\mathrm{BA}}$.

Switching I-frames can be used to gain random access to a stream, as shown in Fig. 1(b). A decoder receiving a stream with a transmission error that affects the P_{A}-frame at $t-1$ can be resynchronized with the SI_{A}-frame at $t+1$. An SI-frame can only be encoded accompanied by an SP-frame.

C. IPTV System Overview

In Fig. 2 we outline a typical IPTV system. The IPTV system consists of three major distinguishable parts, the head-end video server (VS), the clients (C) home apparatus, and the network routers (R) with connecting infrastructure. The type of transport network we address in this paper is a high speed low latency system. Such a network could be a city wide area network. The grey areas in Fig. 2 indicates where the proposed solution affects the system.

D. Scope and Outline

In this paper, we address the issue of reducing redundant information transported over a network due to stream synchronization needs induced by users changing channel and packet loss. Our proposed scheme, Synchronization Frames for Channel Switching (SFCS), separates the main stream from the synchronization stream, transmitting them on two separate multicast groups.

In Sections II, III, and IV we introduce the reader to the application of GOP and SFCS in live-TV over IP environment. Section V includes the network analysis. The simulation environment is presented in Section VI. Finally, the results and conclusions are presented in Sections VII and VIII.

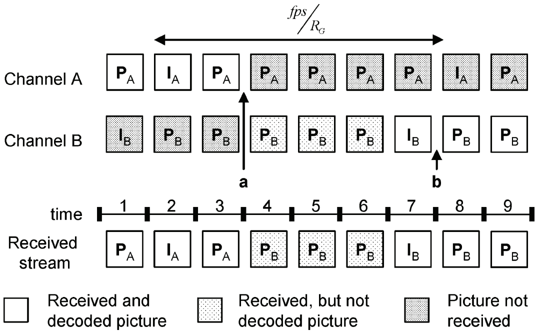


Fig. 3. GOP channel switching.

II. VIDEO TRANSMISSION USING GOP

Group Of Pictures (GOP) organizes the different types of frames in a specific recurring pattern. A GOP typically begins with an I-frame which simplifies stream synchronization followed by P-frames and B-frames. One example of a GOP structure is the widely used IBBPBBPBBPBBI. We denote the distance of the I-frames as the quotient of the number of frames per second (fps) and the number of I-frames per second (R_G), as seen in Fig. 3.

A. GOP Channel Switching

In today's implementations of IPTV systems, it is common to utilize a GOP structure to enable effective synchronization to the transmitted streams (channels). Using a predefined GOP structure is an easy and effective solution to problems associated with stream switching and recovery from information loss.

The example in Fig. 3. illustrates a client performing a stream switch (channel change) in a typical GOP based IPTV system. A client is synchronized to channel A. At a particular time, indicated by (a), the client issues a switch command to channel B. The client sends an IGMP [17] leave to the stream A group and joins the stream B group. The client then starts to receive stream B. Since I-frames act as stream synchronization points for the client decoder, the client waits until an I-frame is received. The received frames are discarded. The waiting time is determined by the number of synchronization points being offered per second (R_G), see Fig. 3. When the I-frame is completely received and decoded, the client is synchronized to the new stream, which is marked by (b).

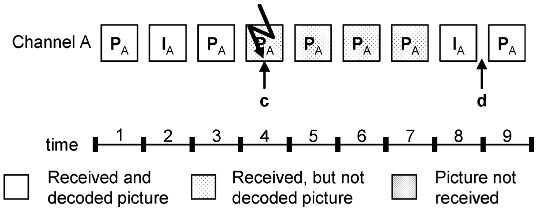


Fig. 4. GOP packet loss.

B. *GOP Information Loss*

The additional redundancy carried by the I-frame only becomes necessary when the client has absolutely no information about the prior frame, i.e. the client is initiating the decoding process of the selected stream. Information loss, as in the case of frame loss, leading to loss of synchronization to the stream is solved by waiting for the next synchronization point (I-frame). This process is clarified in Fig. 4.

A client receiving channel A receives an erroneous frame at time (c), then the decoder waits until a new I-frame is received. During the time period until the I-frame arrives, all intermediate frames are received but not decoded since a suitable reference frame exists. At time (d) the client has received and decoded an I-frame and the decoder is now synchronized to the stream.

III. SYNCHRONIZATION FRAMES FOR CHANNEL SWITCHING

Our proposal, Synchronization Frames for Channel Switching (SFCS), is based on the idea that it is redundant to send intra coded frames at a fixed rate to provide stream resynchronization points due to low rate of channel changing and information loss.

In SFCS, a stream consisting of resynchronization points is separated from the main stream consisting of predictive coded frames only. Receivers (clients) who want to decode the video stream (in this case a TV-channel) must receive both the main stream and the synchronization stream. However, the synchronization stream is only received at the actual synchronization point. The streams are spliced and forwarded to the decoder. The decoded frame at a synchronization point must produce the same result as its counterpart in the main stream to avoid mismatch and drift. One technique capable of performing this is switching pictures (SP-/SI-slices), as described in Section I-B. For SFCS to be more effective than GOP, the size of the encoded SP-frames must be less than the size of the corresponding I-frames. It is also reasonable to assume that the combination of one SI and a corresponding SP-frame is larger than a corresponding I-frame.

Although applications of SP, and SI-frames in video transmission over packet-switched networks have been researched before [15] and are included in the extended profile of the MPEG-4 standard, there has been little or no work invested in the technique applied to an IPTV environment. In this paper, we focus on the implementation of SFCS in a H.264 extended profile enabled environment.

In SFCS, one TV-channel consists of two data streams. One data stream is for the predictive coded frames consisting of inter coded frames (P and SP) which is called the main stream. In addition to the main stream (main), an appurtenant stream consisting of synchronization frames, i.e. SI-frames, is transmitted. This stream is called synchronization stream (sync). Hence, one channel occupies two multicast addresses, one for each stream as shown in Fig. 5.

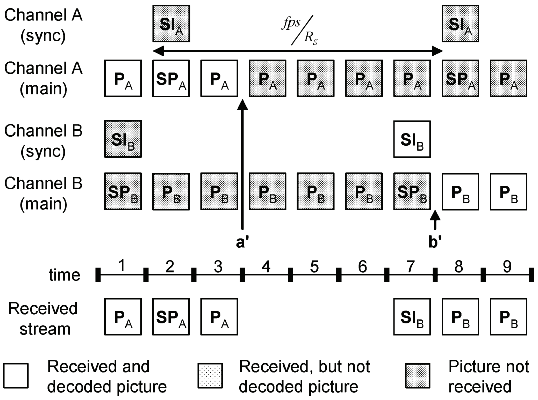


Fig. 5. SFCS channel switching.

A. *SFCS Channel Switching*

In the example illustrated in Fig. 5, a client performs a typical switch from the present channel A to channel B.

The client requests the switch at (a') and departs from the channel A main stream is left and joins the synchronization stream for channel B (Channel B sync). The client awaits traffic on the channel B sync stream. The traffic arrives and is passed to the client decoder. The client is now synchronized to steam B (b'). The client leaves the channel B sync stream and joins the channel B main stream. This method of switching by means of a late joining to the main stream prevents the propagation of unwanted traffic to the client. In a manner similar to the GOP solution, the time between two synchronization points for the SFCS solution is given by the quotient of fps and R_S.

The main benefit of SFCS compared to a GOP scheme is the reduction of link bandwidth for a specified channel due to the lower number of required synchronization frames.

The factors that govern the efficiency of SFCS are; number of receivers, number of selectable channels, channel popularity, frequency of synchronization frame offers, and the time between channel switches.

B. *SFCS Information Loss*

If a client experiences information loss, e.g. one or more data-packets are lost in sequence, the client receiver and decoder will lose synchronization to the stream. An IPTV distribution system, equipped with SFCS, can successfully combat the effects of information loss by offering resynchronization to the stream in the same manner as a channel switch.

Fig. 6 illustrates a single frame loss situation for a client receiving channel A. The receiving client receives an erroneous frame at (c') and immediately leaves the channel A main stream and joins the channel A sync stream. The client awaits traffic on

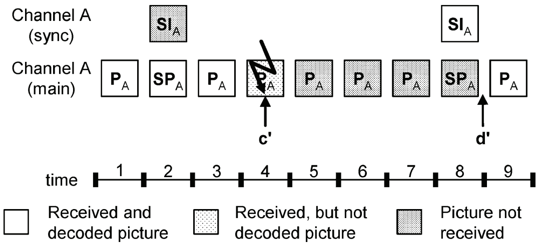


Fig. 6. SFCS packet loss.

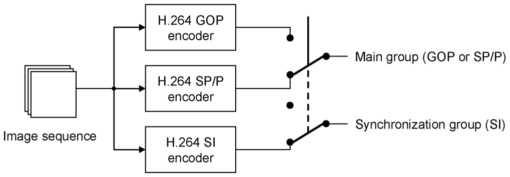


Fig. 7. Server end SFCS-GOP Hybrid system layout for one channel.

the channel A sync stream and when it arrives, the information is forwarded to the decoder and the client is synchronized (d'). The client now leaves the channel A sync stream and rejoins the channel A main stream.

IV. SFCS-GOP HYBRID

Since TV-channels are probably not equally popular and the switching behavior will probably vary over a period of time, it is reasonable to assume that the number of requested synchronizations to a certain stream will vary. If the number of synchronization requests for a specific channel increase enough, the combined load of the main and sync stream for that channel will be greater than the GOP load for the same channel. In this case, the SFCS scheme will perform worse than GOP. We address this by proposing a SFCS-GOP hybrid. The SFCS-GOP hybrid switches between the two transmission techniques depending on which one is more favorable for that channel at that particular time. The SFCS-GOP hybrid solution at the server end for one channel is outlined in Fig. 7. The hybrid switch is controlled by a SFCS-GOP hybrid decision function depending on the number of arriving synchronization requests on that particular channel. The image sequence represents the raw material to be streamed. This sequence is encoded into three different streams, GOP, SP/P, and a SI stream. The GOP and SP/P streams are output on the main channel according to the setting of the SFCS-GOP hybrid state. The SI stream is only outputted on the synchronization stream if the SFCS-GOP hybrid is in the SFCS state. The SFCS-GOP hybrid solution does not further affect the technical solution of the client side of the system.

V. NETWORK BANDWIDTH ANALYSIS

In this section, we make an analytical evaluation of the performance of SFCS compared to GOP in terms of bandwidth in the network carrying the live-TV service. Networks that carry IPTV services are often high speed and low latency networks. Hence, issues connected to latency are not addressed. We consider a network as a set of links connecting nodes as in Fig. 8. A node can either be a router (R) forwarding packets to other nodes, a video server (VS) sending packets to the first router, or a client receiving data from a router. Links between nodes can either be router-to-router link (RRL) or a router-to-client link (RCL).

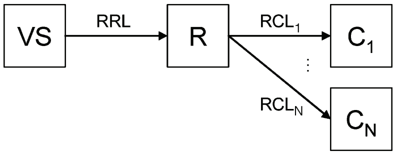


Fig. 8. Network nodes with interconnecting links.

TABLE I
CONSTANTS USED IN GOP AND SFCS EXPRESSIONS

Symbol	Quantity
fps	Frame per second
M	Total number of available channels
N	Total number of clients
d	Average duration between channel changes for one client [s]
N_m	Average number of users receiving channel m
p_m	Probability for a switch to channel m
flp	P-frame loss probability
P_G	Size of one GOP P-frame [bits]
SI	Size of one SI-frame [bits]
SP	Size of one SP-frame [bits]
I	Size of one I-frame [bits]
P_S	Size of one SFCS P-frame [bits]
R_S	SFCS synchronization points per second [1/s]
R_G	GOP synchronization points per second [1/s]
α	Zipf channel popularity distribution exponent.

The RRL carries the video streams between routers, or between the video server and the first router, and the RCL carries selected streams to a single client. Clients are only tuned to one channel at time. All routers are assumed to be multicast enabled. We also assume that information loss equals frame loss. Frames are assumed to be lost randomly and independently. For simplicity, all frame sizes are normalized against the P-frame and Bi-directionally predicted frames (B-frames) are not addressed.

In this investigation, we also address the effect of information loss when using SFCS and GOP respectively. We deliberately limit our theoretical analysis of information loss by only consider the increased number of resynchronization requests caused by the information loss. Information loss probabilities are therefore given in terms of unsuccessful transmission of a whole frame. We also assume that the frame probability loss is proportional to the size of the frame. The probability for a lost SI-frame is therefore higher than the probability for a lost P-frame.

Table I presents the constants used in the bandwidth estimation formulas presented in this section. Table II presents result variables.

A. GOP Channel Switching

The GOP approach is constantly offering resynchronization points (I-frames) and the bandwidth expressions are not affected

TABLE II
RESULT VARIABLES USED IN GOP AND SFCS EXPRESSIONS

Symbol	Quantity
r_I	Size ratio between an I-frame and a P-frame
r_{SP}	Size ratio between an SP-frame and a P-frame
r_{SI}	Size ratio between an SI-frame and a P-frame
δ_{RCL}	Channel changes issued by one client
σ_P	Frame loss probability for one P-frame.
σ_{SP}	Frame loss probability for one SP-frame
σ_{SI}	Frame loss probability for one SI-frame
σ_{SPP}	Frame loss probability for a main channel frame between two synchronization points
γ	Probability for a synchronization frame request caused by information loss for one client
Δ	Synchronization requests introduced by channel changes
Γ	Synchronization requests caused by information loss
Φ	SFCS-GOP hybrid change-over-point
B_{GRCL}	GOP bandwidth expression for one RCL
B_{GRRL}	GOP bandwidth expression for one RRL
B_{SRCL}	SFCS bandwidth expression for one RCL
B_{SRRL}	SFCS bandwidth expression for one RRL
B_{HRCL}	SFCS-GOP hybrid bandwidth expression for one RCL
B_{HRRL}	SFCS-GOP hybrid bandwidth expression for one RRL
W_S	Mean time until next synchronization point for SFCS
W_G	Mean time until next synchronization point for GOP

by either changing the time between switches or altering packet-loss probabilities. I-frames generally consist of more information than P-frames. Therefore, we introduce $r_{I,m} = I_m/P_m$, describing the ratio between the mean size of the I- and P-frames for the channel m.

The GOP bandwidth expression for one RCL link for channel m is then,

$$B_{GRCL,m} = (fps - R_{G,m} + R_{G,m} \cdot r_{I,m})P_m \tag{1}$$

If the number of clients are much lager than the number of available channels $(N \gg M)$ then it is highly probable that all channels traverse that particular RRL. We make this assumption to simplify the bandwidth expression. The GOP bandwidth expression for one channel in the RRL then equals

$$B_{GRRL,m} = (fps - R_{G,m} + R_{G,m} \cdot r_{I,m})P_m. \tag{2}$$

However, if the number of clients are equal or less than the number of available channels, the maximum number of different channels that propagate the particular RRL can not be larger than N. The number of different channels in the RRL depends upon the total number of available channels, the number of clients connected to the router (i.e. RRL), and the channel popularity distribution.

The aggregated bandwidth for all channels in the RRL, given that $N \gg M$, then equals

$$B_{GRRL} = \sum_{m=1}^{M} (fps - R_{G,m} + R_{G,m} \cdot r_{I,m})P_m. \tag{3}$$

The GOP transmission scheme is not generally affected by information loss besides the actual reduction of the information flow. This aspect of information loss is not addressed in the analysis part of this paper.

B. SFCS Transmission Analysis

To make bandwidth estimations, it is of great interest to be able to calculate expected bandwidth consumed in the various links for each channel as well as the aggregated traffic. In a similar manner as the previous section, an analysis of the bandwidth estimations for a system utilizing SFCS is presented.

A client requests the synchronization stream if the client decoder is not synchronized to the main stream. There are two distinguishable events which cause this to happen. One occurs when the client is changing TV-channel. The other occurs when the client loses synchronization to the stream due to information loss.

Since SP- and SI-frames for a channel m consist of more information than the P-frames on average, the two following size-ratios are introduced;

$$r_{SP,m} = SP_m/P_m; \quad r_{SI,m} = SI_m/P_m.$$

The number of synchronization requests in the RCL depends on the rate of the expected number of channel changes issued by the client served $\delta_{RCL} = 1/d$, as well as the synchronization requests caused by information loss, which in this case is denoted as frame loss.

The frame loss probability for one P-frame of channel m equals $\sigma_{P,m}$. It is assumed that the probability for a lost frame is directly dependent on the size of the frame. Consequently, frame loss probabilities for SP/SI-frames for channel m become

$$\sigma_{SP,m} = 1-(1-\sigma_{P,m})^{r_{SP,m}}, \quad \sigma_{SI,m} = 1-(1-\sigma_{P,m})^{r_{SI,m}}.$$

The probability for frame loss in the main data multicast group for channel m during the time between two synchronization opportunities, $\sigma_{SPP,m}$, can be expressed as;

$$\sigma_{SPP,m} = 1 - (1 - \sigma_{SP,m})(1 - \sigma_{P,m})^{\frac{fps - R_{S,m}}{R_{S,m}}}.$$

Thus, the probability for a frame loss in the synchronization data group is $\sigma_{SI,m}$.

As previously described, the traffic in the synchronization channel either originates from channel switching or information loss.

Synchronization stream traffic that originate from information loss is caused either by information loss in the main stream or in the synchronization stream. The probability for a synchronization request originating from a client caused by information loss for the channel m is denoted as $(\gamma_{,m})$. This can then be regarded as the sum of a number of separate events. The pattern can be thought of as, the event of $\sigma_{\mathrm{SPP,m}}$ occurring and the following synchronization frame arrives correctly or the event of $\sigma_{\mathrm{SPP,m}}$ occurring and the synchronization frame being lost ($\sigma_{SI,m}$ occurring) and the next synchronization frame arriving correctly, and so on.

Hence, $\gamma_{,m}$ equals:

$$\gamma_m = \sum_{k=0}^{\infty} \sigma_{SPP,m}(\sigma_{SI,m})^k$$

which equals

$$\gamma_m = \left(1-(1-\sigma_{SP,m})(1-\sigma_{P,m})^{\frac{fps-R_{S,m}}{R_{S,m}}}\right)\sum_{k=0}^{\infty}\sigma_{SI,m}^{k}.$$

For the case where, $0 \leq \sigma_{P,m} < 1$, the following applies

$$\gamma_m = \left\{\begin{array}{ll} 0 & for\ \sigma_{P,m}=0 \\ \frac{1-(1-\sigma_{SP,m})(1-\sigma_{P,m})^{\frac{fps-R_{S,m}}{R_{S,m}}}}{1-\sigma_{SI,m}} & for\ 0<\sigma_{P,m}<1 \end{array}\right\}.$$

The expected mean bandwidth requirement in the RCL can then be written as

$$\begin{aligned} B_{SRCL,m} = {} & (fps - R_{S,m} + r_{SP,m}R_{S,m} \\ & + r_{SI,m}\left(1-(1-\delta_{RCL})(1-\gamma_m)\right))P_m. \end{aligned} \tag{4}$$

The total number of outgoing synchronization requests from the server to the first RLL for each channel can be expressed as the sum of all the requests of the connected clients tuned to the actual channel. Since the synchronization stream is distributed by multicast, there is no issue involving overlapping requests. As for the RCL, the RRL also experiences two types of synchronization traffic. Namely, channel switching and information loss related synchronization traffic. Since the main and synchronization streams are distributed using multicast, the information in the streams is shared by the underlying RCLs. Thus, requests issued during the same timeslot by clients served by the same RRL, share the resulting information flow. The size of the timeslot is defined by R_S.

We define Γ_m as the probability of a synchronization request issued by the clients served by the RRL tuned to the channel m caused by information loss. Hence, $\Gamma_m = 1-(1-\gamma_m)^{N_m}$, where N_m is the number of clients tuned to channel m.

The number of synchronization frame requests introduced by changing to channel m is given by $\Delta_m = 1 - e^{-(N\cdot p_m/R_{S,m}\cdot d)}$, where p_m is the probability for a channel switch to channel m.

The complete bandwidth expression for the first RRL-link assuming $N \gg M$, becomes

$$\begin{aligned} B_{SRRL,m} = {} & (fps - R_{S,m} + r_{SP,m}R_{S,m} \\ & + r_{SI,m}R_{S,m}\left(1-(1-\Gamma_m)(1-\Delta_m)\right))P_m. \end{aligned} \tag{5}$$

For all the channels the expression becomes

$$\begin{aligned} B_{SRRL} = \sum_{m=1}^{M} & (fps - R_{S,m} + r_{SP,m}R_{S,m} \\ & + r_{SI,m}R_{S,m}\left(1-(1-\Gamma_m)(1-\Delta_m)\right))P_m. \end{aligned} \tag{6}$$

For uniform channel popularity distribution $p_m = 1/M$, $N_m = N/M$, together with (6) is

$$\begin{aligned} B_{SRRLu} = \sum_{m=1}^{M} & (fps - R_{S,m} + r_{SP,m}R_{S,m} \\ & + r_{SI,m}R_{S,m}\left(1-(1-\gamma_m)^{\frac{N}{M}}e^{-\frac{N}{R_{S,m}\cdot d\cdot M}}\right)\Big)P_m. \end{aligned} \tag{7}$$

For Zipf [18] distributed channel popularity with the distribution factor α, $p_m = C/m^{\alpha}$, where $C = 1/\sum_{m=1}^{M}(1/m^{\alpha})$, $N_m = (C\cdot N/m^{\alpha})$, with (6) evaluates to

$$\begin{aligned} B_{SRRLz} = \sum_{m=1}^{M} & \Big(fps - R_{S,m} + r_{SP,m}R_{S,m} \\ & + r_{SI,m}R_{S,m}\left(1-(1-\gamma_m)^{N_m}e^{-\frac{p_m}{R_{S,m}\cdot d}}\right)\Big)P_m. \end{aligned} \tag{8}$$

C. SFCS-GOP Hybrid Bandwidth Analysis

The performance of SFCS becomes less than GOP when the number of synchronization requests becomes to high due to channel changes or information loss, or when the number of clients becomes too high. To address this inefficiency problem, an SFCS-GOP hybrid system is proposed. The key issue for optimal performance of such a system is to find the "switch-point", Φ, the instant when to change to GOP instead of SFCS. This can be monitored directly at the VS by measuring the outgoing flows associated to one channel. If the combined traffic of the two SFCS multicast groups associated with one channel is greater than the expected load introduced by a GOP solution, a transition to GOP should be performed. The decision to use GOP instead of SFCS should be monitored continuously and separately for each channel in order to optimize the bandwidth associated with the channel.

The SFCS-GOP hybrid change over point for a specific channel m is calculated by studying $B_{SRRL,m}$ and $B_{GRRL,m}$ for various synchronization frame request situations, as described above. We consider the switch-over case when $B_{SRRL,m}$ and $B_{GRRL,m}$ are equal and solve for the number of synchronization requests per second required to achieve this, this then the SFCS-GOP hybrid threshold for the channel m. Hence,

$$\begin{aligned} & fps - R_{G,m} + R_{G,m}r_{I,m} \\ & \qquad = fps - R_{S,m} + R_{S,m}r_{SP,m} + R_{S,m}r_{SI,m}\Phi_m, \end{aligned}$$

which give

$$\Phi_m = \frac{R_{S,m} - R_{G,m} + R_{G,m}r_{I,m} - R_{S,m}r_{SP,m}}{R_{S,m}r_{SI,m}}.$$

The number of synchronization requests per second φ_m for every channel m, is continuously monitored and analysed at the video server end. The RRL SFCS-GOP hybrid bandwidth expression for the same channel can then be expressed as:

$$B_{HRRL,m} = \left\{\begin{array}{ll} B_{GRRL,m} & if\ \varphi_m > \Phi_m \\ B_{SRRL,m} & if\ \varphi_m \leq \Phi_m \end{array}\right\}.$$

Consequently the RCL bandwidth expression becomes

$$B_{HRCL,m} = \left\{\begin{array}{ll} B_{GRCL,m} & if\ \varphi_m > \Phi_m \\ B_{SRCL,m} & if\ \varphi_m \leq \Phi_m \end{array}\right\}. \tag{9}$$

The complete SFCS-GOP hybrid RRL bandwidth expression is then:

$$B_{HRRL} = \sum_{m=1}^{M} B_{HRRL,m}. \tag{10}$$

The expressions (1)–(10) can be used to calculate the expected bandwidth requirements for the various links in the network.

D. Unsynchronized Frame Reception

Switching to a new channel inevitably causes the receiving equipment to be out of synchronization with the new stream until a synchronization point occurs. The same applies in the event of information loss. The mean expected time until the next synchronization point for the SFCS and GOP for channel m are calculated with

$$W_{S,m} = \frac{fps}{2R_{S,m}} \quad \text{and} \quad W_{G,m} = \frac{fps}{2R_{G,m}}.$$

$W_{S,m}$ and $W_{G,m}$ can be used to indicate the mean time until a channel switch is complete or how long it takes to recover from information loss.

VI. Simulation

To verify the analysis, a simulation environment was created. The simulation environment consists of a custom built simulation environment created in JAVA. The program simulates traffic flows in a simple multicast network as illustrated in Fig. 8. Each client is connected to a router and is tuned to one channel at a time. IGMP messages and related traffic are considered to propagate and take effect immediately in the network.

A. Information Loss

Information loss during one simulation session is simulated by randomly discarding information packets with a fixed probability. In a manner similar to our approach in the analysis section, we consider one P-frame as a unit subject to potential information loss. SI-, I-, and, SP-frames have loss probabilities according to their size compared to a P-frame. All frames are lost independently and all links are assumed to have equal packet-loss characteristics.

B. Channel Switching

Channel switching requests are simulated by a Poisson process with either a uniform or Zipf distributed channel popularity with form factor $\alpha = 1.0$. The average time (d) between channel switches per user is set to 120 seconds instead of 720 seconds to simulate channel switching bursts. The number of active connected clients and available channels are 10000 and 250 respectively, unless otherwise stated. However, although these values are purely assumptions, they represent an estimate of future IPTV solutions.

TABLE III
Frame Size Ratios for GOP and SFCS

Sequence	r_{SI}	r_I	r_{SP}	QPSPSlice
SRC2	7.4106	3.8525	2.0410	27
SRC3	5.4744	2.7485	1.9854	26
SRC4	17.135	9.0228	2.0157	26
SRC5	3.8490	1.8631	1.7752	26
SRC6	2.9782	1.5730	1.6102	26
SRC7	5.1002	2.2034	1.5467	27
SRC8	15.923	7.5595	2.6234	26
SRC9	3.2346	1.6384	1.5057	27
SRC10	7.4902	4.0013	1.9631	27

C. SFCS-GOP Hybrid

In the simulations, the SFCS-GOP hybrid switch-point decisions are made on the fly by analyzing the synchronization requests associated with the channel in question using a 60 second circular buffer with an assumed 2% hysteresis. Hybrid decisions for each channel are made in the video server.

VII. Results

In this section the simulation results are presented and compared with the analysis in Section V.

A. Frame Size Distribution

All channels have different characteristics in terms of their frame-size distribution. This depends in general on the actual content of the channel but also on other factors such as the encoder technology used and the video quality settings used. For this reason, the frame-size distributions of nine video sequences have been investigated, namely the Video Quality Experts Group (VQEG) src2-src10, all being 720 × 576 pixels with 220 frames. The sequences have been encoded using the H.264 JM 10.1 encoder with the SI/SP enhancements by Eric Setton [19]. All sequences are encoded with the QPISlice and QPPSlice parameters set to 30. The QPSPSlice parameter for each sequence was set to match the quality of the corresponding I-frames as closely as possible for each sequence set, see Table III. Corresponding encodings for the GOP and SFCS fall within the 0.5 dB range of each other using $\mathrm{PSNR_Y}$.

Table III displays the size ratios for the investigated video sequences. Individually all the encodings obey the $r_{SI} > r_I > r_{SP}$ relationship. The results in Table III also show that the actual content of the sequence strongly contributes to the individual ratios. Hence, it is reasonable to monitor the frame size ratios continuously and adapt the SFCS-GOP hybrid function accordingly. In all the simulations and results we set r_{SI}, r_I, and r_{SP} to 10, 5, and 2 accordingly. The same is applied to the bandwidth expressions presented in Section V.

B. RCL Bandwidth

SFCS only requests synchronization frames when they are needed. The performance is always better than the GOP case when a separate RCL is considered. This is illustrated in Fig. 9 where we present the results when varying time between channel switches. One client is allowed to switch between two channels and the RCL bandwidth is noted. We also observe

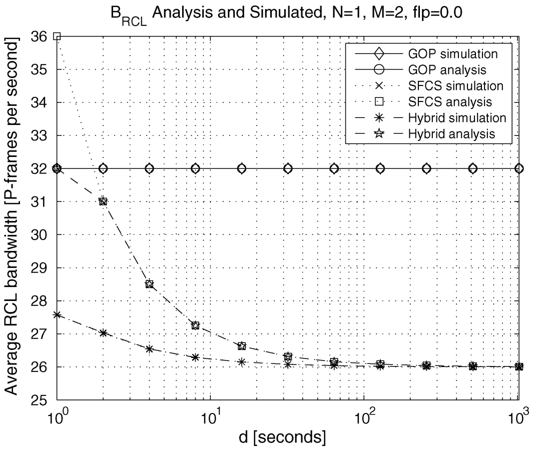


Fig. 9. RCL bandwidth with varying time between channel switches (d) for one client ($N = 1$) and two channels.

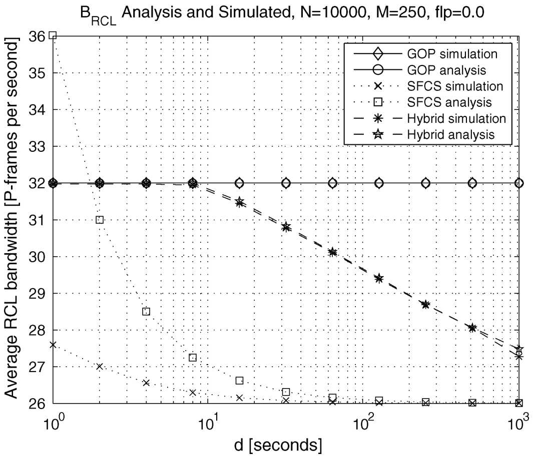


Fig. 10. RCL bandwidth for the investigated transmission modes with varying time between channel switches.

that the simulation for SFCS and SFCS-GOP hybrid differs from the analysis results. This can be explained by the direct leaving of the late joining of the main channel, reducing the traffic delivered to the client.

Also it should be noted that the bandwidth for the hybrid and SFCS transmission modes asymptotically approaches the minimum SFCS bandwidth relatively fast, around 100 seconds between switches. The results in Fig. 10 are based on the average of 10000 RCLs, during which the number of available channels was set to 250. All the clients share the same RRL. As expected, the SFCS-GOP hybrid solution performs worse than the pure SFCS solution when only considering the RCL. This mainly depends on the fact that the SFCS-GOP hybrid decision function takes decisions on the aggregated traffic in the system. The difference between the simulation and analysis of the pure SFCS solution, mainly depends upon the early leave factor.

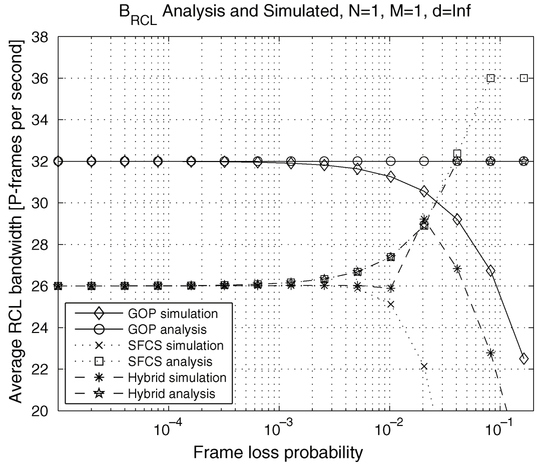


Fig. 11. RCL bandwidth for the investigated transmission modes with varying frame loss probability.

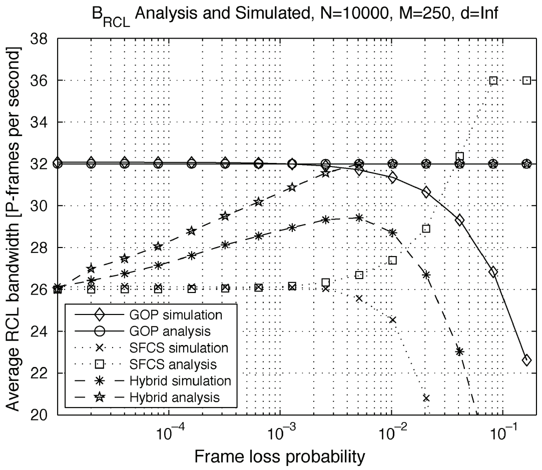


Fig. 12. RCL bandwidth for the investigated transmission modes with varying frame loss probability.

In Figs. 11 and 12, we present the simulation and analysis results with a varying frame loss probability. The scenario with one single client experiencing information loss is displayed in Fig. 11. The hybrid solution is affected by an increased number of connected clients. The reason for this is that the decision if a channel should be in GOP- or SFCS-mode, is made on the traffic from the VS, in this case in the RRL. The increased number of synchronization requests issued by clients experiencing information loss, forces more channels into GOP-mode, this increases the bandwidth used. This is visible in the inclining part of the SFCS-GOP hybrid simulation and analysis curve in Fig. 12. The difference between simulated and analytical results depends mainly on the fact that the analytical model does not take in account the reduction of traffic caused by lost frames. Another contributing factor is the late join effect

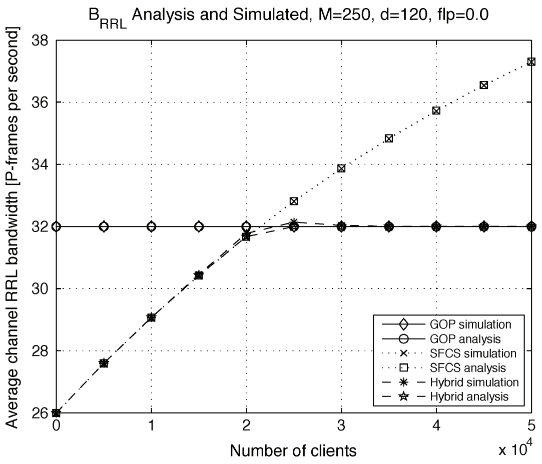


Fig. 13. RRL bandwidth for the investigated methods with a uniform channel popularity distribution with a varying number of connected clients.

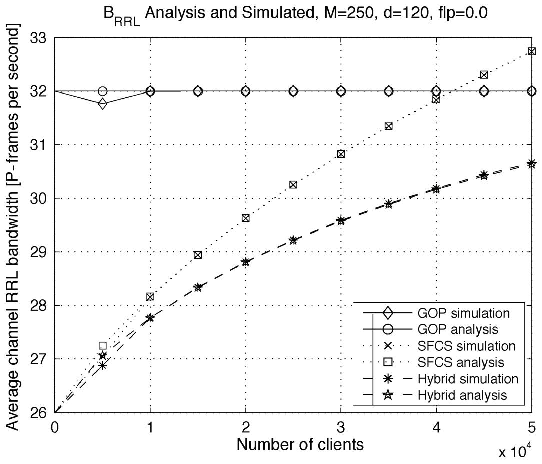


Fig. 14. RRL bandwidth for the investigated methods with a Zipf channel popularity distribution with a varying number of connected clients.

described in Section III-A, which affects performance of the SFCS and SFCS-GOP hybrid transmission methods. The effect of these two becomes apparent when the frame loss probability rises above 1% for Fig. 11. and 0.1% for Fig. 12.

C. RLL Bandwidth

For the RRL, the simple network layout is illustrated in Fig. 8. In Fig. 13 the simulation and analysis results are presented for the average channel bandwidth in the RRL with uniform channel popularity and with no information loss in the system, i.e. flp $=0$. The analysis and simulation results are in agreement.

The average channel RRL bandwidth for Zipf distributed channel popularity is presented in Fig. 14. As previously shown, the simulation and analysis results show near perfect conformance for the investigated set of parameters. A 13% reduction in bandwidth usage can be observed with the SFCS-GOP hybrid solution compared with GOP for 10000 clients and 250 channels.

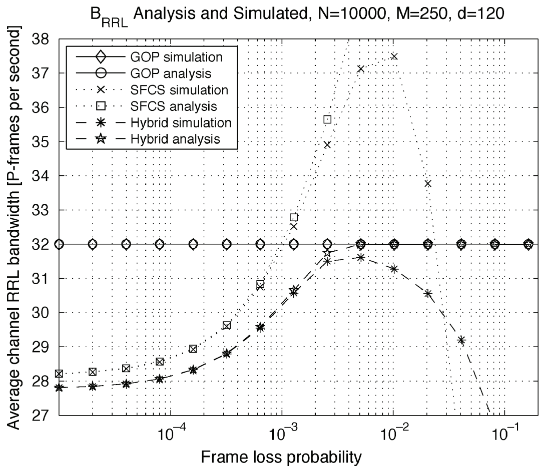


Fig. 15. RRL bandwidth for the investigated methods with a Zipf channel popularity distribution with varying frame loss probability.

The results in Fig. 15 presents the average channel bandwidth for the RRL with 10000 connected clients with 250 selectable channels where the channel popularity is Zipf distributed. It should be noted that for moderate frame loss frequencies, 10^{-3}, the conformance between the simulated and analysis model is more than satisfactory.

VIII. CONCLUSION

In this paper, we have presented an extended use of Synchronization Frames for Channel Switching (SFCS) for H.264 based live-TV distribution systems, which forms part of future IPTV solutions. The analysis section includes complete bandwidth estimation expressions for both router client links (RCLs) as well as router to router links (RRLs). These expressions serve as tools for estimating bandwidth requirements for an H.264 based live-TV service in an IPTV environment. In addition to the effects introduced by channel switching behavior, effects caused by information loss introduced by unreliable transmission channels are included. We also present an effective SFCS-GOP hybrid solution to address the infectivity issues of SFCS. We also include a schematic server system layout and methods to calculate the SFCS-GOP switch-over points for the SFCS-GOP hybrid. Our results show that the SFCS-GOP hybrid solution significantly increases bandwidth utilization in terms of average channel bandwidth on RRLs. This holds true even for large number of users and a relatively small set of selectable channels when compared to a traditional GOP based live-TV transmission system. In addition, the SFCS-GOP hybrid always performs better or equal to a traditional GOP based system. The simulation results show that for a RRL serving 10000 clients which offers 250 channels, a reduction of 13% in bandwidth compared to similar GOP solution is observed. The bandwidth gains produced by SFCS and the SFCS-GOP hybrid solution can be used to adaptively reduce the waiting time between channel switches by increasing the frequency of synchronization points offered per second. Since the merging of synchronization and main stream can be done before the decoder, the actual implementation of the decoder is not an issue, as long as the requirements for SFCS are met.

The efficiency of SFCS is dictated by; the number of receivers, the number of selectable channels, the channel popularity, the frequency of synchronization frame offers, the duration between channel switches, and general network conditions affecting information loss.

A. *Future Work*

Most of the compression schemes in use today do not have SP/SI-frames functionality. Although it is part of the extended profile in the H.264 standard, it suffers from poor support by hardware solutions. Therefore, it is of interest to examine the possibilities to implement a variant of SFCS which can function in conjunction with a system not supporting SI/SP-frames. The problem of drift introduced by synchronization-stream frame mismatch to the corresponding main-stream frame can be addressed by means of further spaced intra coded frames in the main stream and variable quality synchronization stream frames.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their thoughtful comments and suggestions in improving this paper.

REFERENCES

[1] F. Kozamernik and L. Vermaele, “Will broadband TV shape the future of broadcasting,” *EBU Technical Review*, April 2005.
[2] J. Walko, “I love my IPTV,” *IEE Communications Engineer*, vol. 3, no. 6, pp. 16–19, Dec./Jan. 2005/06.
[3] S. Cherry, “The battle for broadband,” *IEEE Spectrum*, Jan. 2005.
[4] Post och Telestyrelsen, Bredband i Sverige 2005 PTS-ER-2005:24, 1650-9862, June 2005.
[5] Informa Telecoms & Media, “IPTV: a global analysis,” *Informa Telecoms & Media*, Aug. 2005.
[6] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2, International Organization for Standards, 1996.
[7] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Systems*, ISO/IEC 13818-1, International Organization for Standards, 1996.
[8] A. J. Stienstra, “Technologies for DVB services on the Internet,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 228–236, Jan. 2006.
[9] FASTWEB [Online]. Available: http://www.fastweb.it
[10] FastTV [Online]. Available: http://www.fasttv.net
[11] M. Brennan and M. Syn, “Television viewing behaviour during commercial breaks,” in *Australian & New Zealand Marketing Academy Conference (Anzmac 2001)*, Dec. 2001.
[12] J. M. Boyce and A. M. Tourapis, “Fast efficient channel change [set-top box applications],” in *International Conference on Consumer Electronics (ICCE 2005)*, Jan. 2005, pp. 1–2.
[13] U. Jennehag and T. Zhang, “Increasing bandwidth utilization in next generation IPTV networks,” in *IEEE International Conference on Image Processing (ICIP 2004)*, Oct. 2004.
[14] M. Karczewicz and R. Kurceren, “The SP- and SI-frames design for H.264/AVC,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, July 2003.
[15] N. Färber, E. Steinbach, and B. Girod, “Robust H.263 compatible transmission for mobile video server access,” in *First International Workshop on Wireless Image/Video Communications*, Sep. 1996.
[16] M. Kalman and B. Girod, “Rate-distortion optimized streaming of video with multiple independent encodings,” in *IEEE International Conference on Image Processing (ICIP 2004)*, Oct. 2004.
[17] W. Fenner, “Internet group management protocol, version 2,” Network Working Group, Request for Comments: 2236, Nov. 1997.
[18] G. Zipf, *Human Behaviour and the Principle of Least Effort*. Cambridge, MA: Addison-Wesley, 1949.
[19] SP Frame Encoder/Decoder [Online]. Available: http://www.stanford.edu/~esetton/H264_2.htm

Ulf Jennehag received the M.Sc. degree from Mid Sweden University, Sweden, in 2000, and the Licentiate of Engineering from Royal Institute of Technology (KTH), Sweden, in 2005.

He is now pursuing his doctoral degree at Mid Sweden University where he also is a researcher within the MUCOM lab. His research interest is in digital video transmission and video coding.

Tingting Zhang (M’94–S’05) received the B.Sc. degree, and M.Sc. degree in Computer Science and Engineering from Fudan University, Shanghai, China, in 1982 and 1984 respectively.

During 1985–1987, she worked as a lecturer at Fudan University. In 1987 she became a research student in Linköping University, and in 1993 she received the Ph.D. in Computer Science and Engineering. Meanwhile she worked as a university lecturer for one year at University College of Gävle/Sandviken. Since 1993, she has been a university lecturer in Computer Science at Mid-Sweden University, where she is also the leader of the MUCOM lab. She is interested distributed systems, digital video transmission and sensor networks.

Stefan Pettersson (M’05) received his B.Sc. in Electrical Engineering and his B.Sc. in Computer Science from Mid Sweden University in 1995. In 2004 he received his Ph.D. in Radio Communication Systems from the Royal Institute of Technology.

Since 2004, he has been employed as a university lecturer at Mid Sweden University where he is also a senior researcher within the MUCOM lab. His research interest is in digital video transmission and wireless sensor networks.

Paper V

On Synchronization Frames for Channel Switching in a GOP-based IPTV Environment

Ulf Jennehag and Stefan Pettersson

Published in
Proceedings of the fifth IEEE Consumer Communications & Networking Conference (CCNC 2008),
Las Vegas NV, USA, January 2008.

On Synchronization Frames for Channel Switching in a GOP-based IPTV Environment

Ulf Jennehag[†] and Stefan Pettersson
Department of Information Technology
Mid Sweden University
Sundsvall, Sweden
Email: Ulf.Jennehag@miun.se, Stefan.Pettersson@miun.se

***Abstract*— Traditionally, Group of Pictures (GOP) is used to solve IPTV channel tune-in issues. However, channel changes issued by a TV viewer are quite rare which indicates it is redundant to solve channel tune-in with GOP. In this paper we extend our earlier work on Synchronization Frames for Channel Switching (SFCS) for reducing bandwidth utilization in IPTV networks to include encoding/decoding standards not supporting SI/SP-frames. We propose modifications to SFCS to work with MPEG-2. Channel tune-in quality and synchronization stream generation is also addressed. We provide bandwidth estimation expressions which are verified with an experimental simulation. Results show that SFCS adapted to MPEG-2 can reduce the bandwidth utilization with 9% compared to a traditional GOP system with equal steady state quality.**

I. INTRODUCTION

In recent years, the growing interest in Internet Protocol Television (IPTV) [1] [2] has grown rapidly and is now available for over 100 million households all over the world. IPTV is generally considered to specify two types of services, live-TV distribution via multicast and Video on Demand (VoD). Digital video content is commonly distributed via MPEG-2 Transport Stream (TS)[3]. A single MPEG-2 Program Stream (PS) is extracted and multiplexed into a new TS which is distributed over IP with multicast.

Traditionally, the video in IPTV streams is encoded using a predefined group of pictures (GOP). The GOP determines which picture type to encode at a certain time instance. A commonly used GOP is the well known ...IBBPBBPBBPBBI... The rather high frequency of intra coded frames (I-frames) allows the receiver to rapidly tune in to the stream. In general, an I-frame requires more information than a predictive coded frame (P-frame or B-frame). With this in mind it makes sense to keep then number of I-frames as low as possible to either reduce the needed bandwidth or increase the quality of the coded video with maintained bandwidth. This can be beneficial to IPTV deployments where the last-mile links, are bandwidth constrained. Brennan and Syn [4] show that the average channel switching rate is 4.9 switches per hour during programs and 7.2 switches per hour during commercials, averaging to a 5.4 switches per hour. This indicates that the number of I-frames needed for stream synchronization due to channel switches is quite low. The perceived quality of the channel switch is also important. A study by Kooij et al [5] indicate that user acceptance for long channel switching times is quite low. However, zapping times below 0.5 seconds are perceived as instantaneous. Whereas channel switching times above 1.5 seconds are perceived as annoying.

In this paper, we will extend the idea of using Synchronization Frames for Channel Switching (SFCS) [6][7] to increase bandwidth utilization and channel tune-in speed. In our previous research, we have focused on increasing the transmission efficiency of H.264 SI-/SP-frames enabled IPTV environment. However, many current IPTV deployment still rely on MPEG-2 for both for video compression [8] and transport [3]. Hence it makes sense to suggest improvements in transmission efficiency to such systems. Similar approaches have also been addressed by Boyce and Tourapis [9].

We will modify and adapt our SFCS approach so it works in a Group of Pictures (GOP) transmission environment and present mathematical expressions for bandwidth estimation. The bandwidth estimation can be used to calculate the expected mean bandwidth usage for an IPTV system. We will also address the expected tune-in quality problem, i.e. the expected difference in the decoded output of the tune-in picture compared to its corresponding main video stream picture. This difference may result in visible impairments. In addition we will address the problem with overlapping I-frames in tune-in and main video streams. We present simulation results regarding tune-in stream quality and propose a method of tune-in stream generation. It is important to note that the tune-in time for audio content also affects the perceived quality of the channel switch. However, in this paper we will only address the video part of the IPTV stream.

A. Outline of the paper

In the next section, we introduce the reader to GOP and SFCS channel switching plus network performance issues. In section III, we present bandwidth estimation expressions for the proposed system. Section IV describes the used simulation environment and the experiment setup with related assumptions. The results are presented in section V and the conclusions in section VI.

[†] The author is also affiliated with Acreo AB, Photonics, Box 1053, SE-82412 Hudiksvall, Sweden. The work was financed in part by the Regional Fund of the European Union and the County Administrative Board of Västernorrland Sweden.

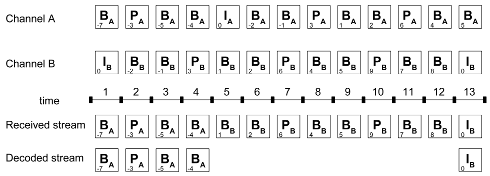


Fig. 1. GOP channel switch example

II. CHANNEL SWITCHING

To be able to evaluate the performance of our SFCS system, we compare with a traditional GOP-based transmission system. Below, we explain the features of the compared transmission systems in two examples describing a channel change situation. The GOP and SFCS channel switch situations are illustrated in Fig.1 and Fig.2. The time mentioned in the example texts indicates the transmission and decoding order of the frames in the streams. The number in the individual frames indicates the presentation order. All streams are distributed with multicast.

A. Group of Pictures

A GOP-transmission system utilizes I-frames as stream synchronization points. Hence, the GOP structure will determine the tune-in time and the bandwidth requirements. A channel switch situation in the GOP-transmission system is illustrated in Fig.1. The client requests a channel switch at time instance 5 (t5). The client then leaves channel A stream and immediately joins the channel B stream (t5). The client receives data from channel B and the decoder waits for the next I-frame in order to synchronize with the stream. At (t13) the decoder is synchronized to the new stream.

B. Synchronization Frames for Channel Switching

Synchronization Frames for Channel Switching is based on the idea that it is inefficient to send intra coded frames at a fixed frequency to provide stream resynchronization points when the switching rate is low. With SFCS, a stream consisting of stream synchronization points (I-frames) is separated from the main video stream which consists of a normal GOP structure with a large I-frame distance. Clients who want to decode a video stream, i.e a TV-channel, must receive both the main video stream (*main*) and the synchronization stream (*sync*). However, the synchronization stream is only joined and received at the actual synchronization point and is later left. The streams are then spliced and decoded. The following example, illustrated in Fig.2, will explain a typical SFCS channel switch situation. The client requests the channel switch at time instance 5 (t5), the A *main* multicast stream is immediately left (t5) and the *sync* stream for channel B is joined (t5). The client waits until traffic arrives and receives the I-frame (t7). The *sync* stream is left when the complete I-frame is received. The *main* stream of channel B is now joined and the I-frame is decoded (t7). The client is now synchronized to channel B.

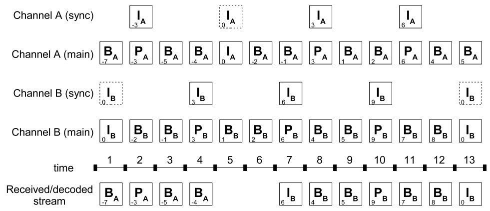


Fig. 2. SFCS channel switch example

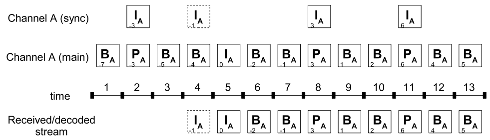


Fig. 3. Suggestion for removing the redundant I-frame in the *sync* stream.

C. Redundant Synchronization Data

As shown in Fig.2, there exist situations where I-frames overlap in the *sync* and *main*, e.g. stream A (t=5). For example, a client joins the *sync* stream and receives an I-frame, but at the same time there is an I-frame in the *main* stream which is a better choice since it provides a direct synchronization to the *main* stream with no risk for additional drift. We propose a solution by offsetting the corresponding *sync* I-frame by one time unit. This special *sync* I-frame represent a low complexity image, i.e black picture, which is forwarded to the decoder followed by the *main* stream which immediately starts with an I-frame. The whole process is exemplified in Fig.3. This method reduces the drift and the network load compared to sending a full quality I-frame in the *sync* stream. The first two B-frames received can be discarded since it is not possible to perform a correct decoding of these due to the missing preceding reference frame.

III. NETWORK ANALYSIS

In this chapter, the network analysis is addressed. We define variables and state bandwidth estimation expressions for both SFCS and GOP transmission techniques using two different channel popularity distributions. The presented bandwidth estimation expressions can be used to calculate expected mean bandwidth usage for an IPTV system utilizing SFCS, as well as GOP. Networks that carry IPTV services are often high-speed with low latency. In this paper we consider a network as a set of links connecting nodes. The nodes can either be a client node (C) or a router node (R). Client nodes issues channel changes and receives traffic. Router nodes routes traffic to clients or other routers. There is also one video server (VS) acting as stream origin for all multicast in the network. The VS also have router capability. The network layout is displayed in Fig.4. Links connecting routers to other routers are called router-to-router links (RRL). Links connecting routers and clients are called router-to-client links (RCL). All routers in the

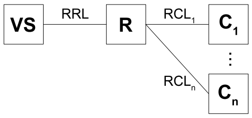


Fig. 4. Network layout

TABLE I
GOP AND SFCS PARAMETERS

Symbol	Description
fps	frames per second
M	number of available channels
N	total number of clients
d	average duration between channel changes
p_m	probability for a switch to channel m
N_m	number of clients tuned to channel m
P_{GOP}	Average size of a P-frame in GOP [bits]
R_{ISM}	I-frame rate SFCS main
R_{PSM}	P-frame rate SFCS main
R_{BSM}	B-frame rate SFCS main
R_{ISS}	I-frame rate SFCS sync
R_{IG}	I-frame rate GOP
R_{PG}	P-frame rate GOP
R_{BG}	B-frame rate GOP
r_{ISS}	(Average I-frame size in *sync*)/P_{GOP}
r_{ISM}	(Average I-frame size in *main*)/P_{GOP}
r_{PSM}	(Average P-frame size in *main*)/P_{GOP}
r_{BSM}	(Average B-frame size in *main*)/P_{GOP}
r_{IG}	(Average I-frame size in GOP)/P_{GOP}
r_{BG}	(Average B-frame size in GOP)/P_{GOP}

network are multicast enabled. The number of synchronization requests caused by clients switching channels each time unit is modelled by a poisson process with a number of channel switches every time unit. A time unit is $1/fps$ second.

The bandwidth estimation expression for the RCL using GOP transmission is

$$B_{GRCL} = (r_{BG}R_{BG} + R_{PG} + r_{IG}R_{IG})\,P_{GOP}. \tag{1}$$

Bandwidth estimation expression for the RRL using GOP transmission mode for one channel evaluates to

$$B_{GRRL,m} = (r_{BG}R_{BG} + R_{PG} + r_{IG}R_{IG})\,P_{GOP}. \tag{2}$$

And for all the channels in the RRL, assuming $N \gg M$, evaluates to

$$B_{GRRL} = \sum_{m=1}^{M} B_{GRRL,m}. \tag{3}$$

In similar manner as in (1)-(3) we derive the corresponding bandwidth estimation expressions for the SFCS transmission mode. We define the *main* stream channel bandwidth as

$$B_{SM} = (r_{BSM}R_{BSM} + r_{PSM}R_{PSM} + r_{ISM}R_{ISM})\,P_{GOP}. \tag{4}$$

Hence, the SFCS RCL bandwidth estimation expression becomes

$$B_{SRCL} = B_{SM} + \tfrac{r_{ISS}R_{ISS}P_{GOP}}{d}. \tag{5}$$

The number of synchronization opportunities per second is defined as

$$R_{IS} = R_{ISS} + R_{ISM}. \tag{6}$$

The RRL bandwidth expression then follows as

$$B_{SRRL,m} = B_{SM} + r_{ISS}R_{ISS}P_{GOP}\left(1 - e^{-\frac{Np_m}{dR_{IS}}}\right). \tag{7}$$

And the aggregated SFCS RRL estimated bandwidth expression is

$$B_{SRRL} = \sum_{m=1}^{M} B_{SRRL,m}. \tag{8}$$

The p_m indicates the channel popularity. We consider two channel popularity distributions, uniform and Zipf [10]. For uniform channel popularity distribution we get

$$p_m = \frac{1}{M} \tag{9}$$

Combining (8) and (9) gives

$$\begin{aligned} B_{SRRLu} &= MB_{SM} \\ &+ Mr_{ISS}R_{ISS}P_{GOP}\left(1 - e^{-\frac{N}{dMR_{IS}}}\right). \end{aligned} \tag{10}$$

For Zipf distributed channel popularity with form factor α then p_m becomes

$$p_m = \frac{1}{m^{\alpha}C} \; where \; C = \sum_{c=1}^{M} \frac{1}{c^{\alpha}} \tag{11}$$

which when combined with (8) gives

$$\begin{aligned} B_{SRRLz} &= MB_{SM} \\ &+ r_{ISS}R_{ISS}P_{GOP}\sum_{m=1}^{M}\left(1 - e^{-\frac{N}{dm^{\alpha}CR_{IS}}}\right). \end{aligned} \tag{12}$$

IV. SIMULATION ENVIRONMENT AND SETUP

The two investigated transmission techniques are simulated in a Java program consisting of a virtual network with clients (C_1-C_n) and a video server (VS), i.e a head-end, as a single stream origin point. Single clients use a random process to decide when to switch channel. Channel popularity is determined by a channel popularity distribution, uniform or Zipf distributed. Clients can issue a channel change every time unit which is equal to fps number per second. All the links between nodes in the virtual network is considered to be error free. Multicast group management information is assumed to be delivered error free and take effect immediately. All the reported simulation results in section V and the corresponding results derived from the bandwidth estimation expressions in section III utilize the same parameters. A schematic picture of the network used in the simulations is presented in Fig.4. Luminance PSNR [11] was used as a quality measure of the decoded sequences.

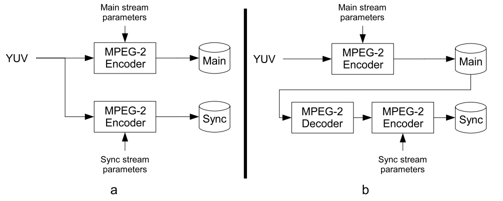


Fig. 5. Block scheme of the *main* encoding approach

TABLE II

EXPERIMENTAL FRAME SIZE RATIOS

Symbol	Value	Symbol	Value	Symbol	Value
R_{ISS}	1.5	R_{ISM}	0.5	R_{PSM}	7.5
R_{BSM}	16	R_{IG}	2	R_{PG}	6
R_{BG}	16	r_{ISS}	1.2925	r_{ISM}	1.8752
r_{PSM}	0.9778	r_{BSM}	0.3672	r_{IG}	1.9319
r_{PG}	1.0	r_{BG}	0.3885	fps	24
α	0.95	M	100	d	120

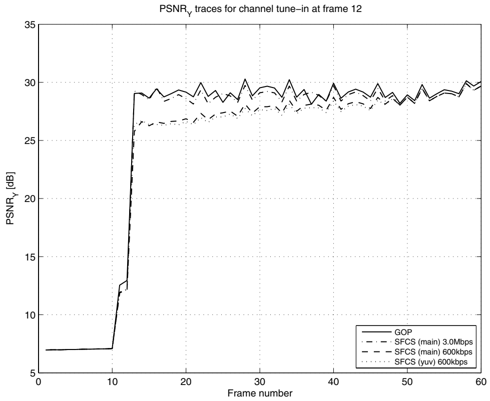


Fig. 6. PSNR traces for the investigated tune-in scenarios

The streams where encoded/decoded with MPEG Software Simulation Group (MSSG) MPEG-2 encoder/decoder [12] v1.2. The "mobile and calendar" video clip was used as source material for the conducted experiments. The *main* and GOP stream was encoded so the resulting PSNR was as equal as possible. The *sync* stream was encoded with varying quality and encoding approaches. Channel tune-in performance is measured by luminance PSNR of a decoded spliced (*sync*+*main*) sequence.

A. Encoding Synchronization Stream

I theory, the *sync* stream should contain coded frames that exactly matches its counterparts in the *main* stream. We propose two approaches for encoding the *sync* stream. The first is to encode the *sync* stream using the same source sequence as the *main* stream, this approach is called the YUV encoding and is illustrated in Fig.5 (a). As mentioned earlier, it is crucial that the decoded result of the *synch* stream I-frame$_n$ is equal to the corresponding decoded *main* stream P-frame$_n$ in order to minimize drift. We therefore propose a second encoding approach taking this matter into account. Instead of using the same source material as the *main* stream the decoded representation of the *main* stream is used, this enables the *sync* stream generation to achieve an I-frame with minimal difference to the corresponding decoded *main* stream P-frame. The described method is referred to as *main* encoding, illustrated in Fig.5 (b). The performance of both the encoding approaches are compared and the results are presented in section V-B.

V. RESULTS

In this section we present the results from the frame-size investigation and the simulated bandwidth consumption of the simulation environment compared to the results from the bandwidth estimation expressions presented in section III.

A. Frame Size Ratios

We use the values in Table II for our simulations and for the bandwidth estimation expressions presented in section III. The GOP stream was encoded at 5.0 Mbps and yielded a luminance PSNR of 29.80 dB for the whole sequence consisting of 336 frames. The GOP size was set to 12 frames at 24 fps. The SFCS *main* stream was encoded at 4.4 Mbps and yielded a luminance PSNR of 29.84 dB. The *sync* stream used in our simulations was encoded at 600 kbps with approximately 1.5 fps, i.e. every 12th frame excluding overlapping I-frames in the *main* stream as described in section II-C.

B. Channel Tune-in Performance

In this section we present the results from the investigation of channel tune-in. A channel change was simulated by splicing one *sync* stream I-frame with the corresponding P-frame in the *main* stream for three different *sync* stream settings. The resulting tune-in stream was decoded and PSNR was compared with the corresponding GOP, illustrated in Fig.6. The channel switch was issued before frame 12 in both cases. The first two received B-frames were decoded, explaining the glitch on the rising edge of the curves. Between frame 12 to 48 the average luminance PSNR difference between the GOP and SFCS tune in where 1.56 dB.

The average luminance PSNR difference between the *main* encoding and the YUV encoding approach is 0.18 dB favoring the *main* encoding approach. Even though the *main* encoding approach looks more complex it has one big advantage over the YUV encoding approach by not requiring the image sequence the *main* stream is constructed from. The *sync* stream can thus easily be constructed.

C. RCL Bandwidth

Since an SFCS enabled client only request synchronization frames when changing a channel, the SFCS always perform

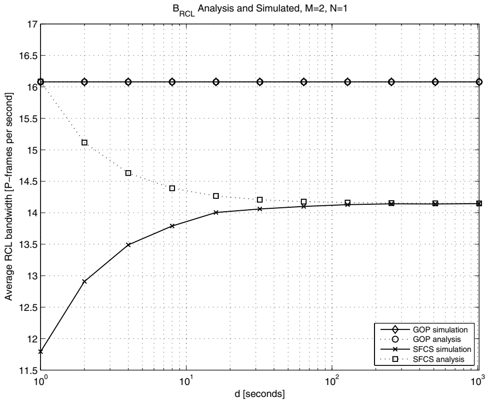


Fig. 7. B_{RCL} for one client switching between two channels with varying d.

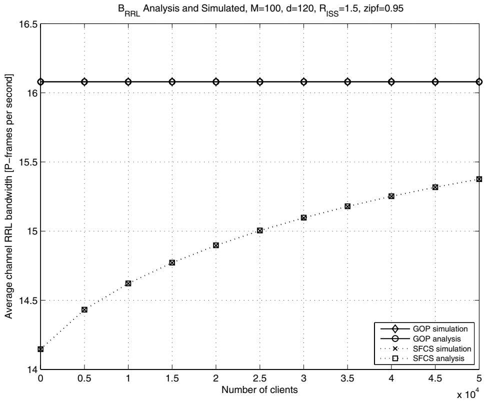


Fig. 8. B_{RRL} with varying number of connected clients.

better than a similar GOP system when considering the traffic in a single RCL. The simulation results compared with analysis results for one single RCL is shown in Fig.7. The difference between simulated and analytical results mainly depends upon the reduction in traffic that the leave of the previous channel causes. There is a 12% reduction of the steady-state bandwidth in the RCL when using the SFCS system compared to GOP.

D. RRL Bandwidth

The network studied is shown in Fig.4, the parameters are set to the values presented in tableII and the number of connected clients are varied from 0 to 50000. The simulation and analysis result for B_{RRL} per channel is shown in Fig.8 for both SFCS and GOP system. A 9% reduction in bandwidth for 10000 clients and 100 channels can be observed for the SFCS transmission system compared to GOP. The conformance between simulated and analytical results is very good.

VI. CONCLUSION

In this paper we have presented an extension to SFCS to be compatible with encoding/decoding systems that do not support SI/SP-frames. In our investigations we have used MPEG-2. Further, we have proposed a method to address the potential problem with overlapping I-frames in *sync* and *main* stream. The tune-in quality was addressed, we have presented a typical tune-in situation and the PSNR quality difference between SFCS and GOP for the tune in time is about 1.6 dB average for the given parameters. The difference is considered to acceptable considering the time of exposure. We have also presented a method to construct the *sync* stream that yields slightly better performance with little increased complexity than the YUV encoding method. We presented bandwidth estimation expressions to easily compute SFCS and GOP bandwidth requirements in IPTV networks. The conformance between the simulated and the analysis results is very good. SFCS has shown to be a good approach to lower the bandwidth utilization in a network distributing IPTV services. Traffic in the RRL is reduced with about 9% compared to similar GOP system with maintained quality for the given variables. In the RCL the steady state bandwidth is reduced with 12% for the presented variables.

REFERENCES

[1] P. Pfeffer, "IPTV: Technology and development predictions," *Fiber and Integrated Optics*, vol. 25, no. 5, pp. 325–246, September-October 2006.
[2] A. Yarali and A. Cherry, "Internet protocol television (IPTV)," in *TENCON 2005 2005 IEEE Region 10*, November 2005, pp. 1 – 6.
[3] *Information Technology - Generic Coding of Moving Pictures and Associated Audio: Systems Rec. H.222.0*, International Organisation for Standardisation Std. ISO/IEC 13 818-1, Rev. E, November 1994.
[4] M. Brenann and M. Syn, "Television viewing behaviour during commercial breaks," in *Proceedings of the 2001 Australian & New Zealand Marketing Academy (ANZMAC2001) Conference*, dec "3–5," 2001.
[5] R. Kooij, K. Ahmed, and K. Brunnström, "Percived quality of channel zapping," in *Proceedings of th efith IAESTED International Conference Communication Systems and Networks*, August 28-30 2006, pp. 155–158.
[6] U. Jennehag and T. Zhang, "Increasing bandwidth utilization in next generation IPTV networks," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 3, October 2004, pp. 2075 – 2078.
[7] U. Jennehag, T. Zhang, and S. Pettersson, "Improving Transmission Efficiency in H.264 Based IPTV Systems," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 69–78, March 2007.
[8] *Information Technology - Generic Coding of Moving Pictures and Associated Audio: Video Rec. H.262*, International Organisation for Standardisation Std. ISO/IEC 13 818-2, Rev. E, 1995.
[9] J. M. Boyce and A. M. Tourapis, "Fast efficient channel change [set-top box applications]," in *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, jan 2005, pp. 1–2.
[10] G. K. Zipf, *Human behavior and the principle of least effort*. Addison-Wesly Press, Inc., 1949.
[11] *Digital Transport of One-Way Video Signals - Parameters for Objective Performance Assessment*, American National Standards Institute Std. T1.801.03-1996, February 1996.
[12] MPEG Software Simulation Group. (1996, 19 July) MPEG-2 Encoder/Decoder V1.2. [Online]. Available: ftp://ftp.sunet.se/pub/multimedia/graphics/ImageMagick/delegates/mpeg2vidcodec_v12.tar.gz