Packet Loss Effects on the Quality of MPEG-2 Video Transported Over IP Networks

by

Daniel Chiu

B.Sc., University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

we accept this thesis as conforming
to the required standard

/

The University of British Columbia

December 1999

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMPUTER SCIENCE

The University of British Columbia
Vancouver, Canada

Date 18 FEBRUARY 2000

Abstract

In this thesis, we examine the effects of packet loss on the quality of the MPEG-2 video transported over an IP network. We performed experiments in a video-on-demand testbed network using three constant bit rate MPEG-2 video elementary streams of differing activity levels. We assessed the video quality using the objective peak signal to noise ratio (PSNR) metric and the subjective mean opinion score (MOS) metric. Our results confirmed that the objective PSNR video quality metric is poorly correlated with subjective quality assessments. We found that the activity level of the video influences the subjective MOS video quality assessments but not the objective PSNR quality measurements. The difference in the perceived video quality between the activity streams was found to be attributed to characteristics in the human visual system. In particular, a high temporal frequency decreases the sensitivity of the visual system. We found that slice loss is linearly correlated with packet loss while picture header loss is poorly correlated with packet loss. We also found that slice loss is the dominant factor contributing to the degradation in video quality rather than picture header loss. We found that low packet loss rates translate into much higher frame error rates due to propagation of errors. Finally, we investigated the effectiveness of forward error correction (FEC) using Reed-Solomon coding on the video quality. We found that FEC can increase a stream's packet loss

rate tolerance significantly with a small increase in overhead.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank Prof. Gerald Neufeld and Prof. Mabo Ito for their guidance and support both academically and financially throughout the course of this work. I would like to thank Mark McCutcheon, our research associate in the TEVIA project, for all his constructive suggestions, comments, and help. I would also like to thank my partner Norton Cai for his work on the video quality measurement tools as well as the endless discussions and invaluable insight on the subject.

Finally, I would like to thank my family and friends for their love, trust, support, and encouragement throughout all these years. I am grateful to you.

DANIEL CHIU

The University of British Columbia

December 1999

To my parents to whom I owe every success that I achieve.

Chapter 1

Introduction

Multimedia services involving the delivery of digital video and audio streams over networks represent the future of conventional home entertainment, encompassing cable and satellite television programming as well as video-on-demand, video games, and other interactive services. This evolution is enabled by the rapid deployment of high-bandwidth connections, such as ADSL and cable modem technology, to the home. The ubiquity of the Internet and the continual increase in computing power of the desktop computer together with the availability of relatively inexpensive MPEG-2 decoder plug-in cards have made MPEG-2 based, high quality video communications an interesting possibility.

With digital video likely to become the dominant type of multimedia for on-demand services which will be offered to the consumer for a fee, it will be imperative to provide them with acceptable quality of service. However, it is in the interest of the service provider to make efficient use of the available network bandwidth. An overloaded or over-subscribed network will result in reduced quality of service that may no longer be acceptable to the paying consumer. Thus, there is a need

to determine the network conditions that will optimally use the network bandwidth but at the same still provide acceptable video quality.

This thesis deals with the transport of MPEG-2 video in an IP-based network. At present, the preferred encoding technique for digital broadcast video is MPEG-2. MPEG-2 exploits both spatial and temporal redundancies enabling compression ratios up to 200:1. It can encode both video and audio sources to almost any level of quality, from VCR- to HDTV-quality.

Many previous studies have investigated the transport of MPEG video over ATM. Few have considered the transport of MPEG video over IP due to its lack of quality of service (QoS). Researchers have examined the various aspects involved in the transport of MPEG video over ATM such as system design, performance requirements, error control and concealment. Quality of service and system design issues for supporting MPEG-2 video over ATM have been examined [49, 56]. In [36], Lee et al. propose a simple feedback cell drop scheme that provides better performance when supporting real-time MPEG-1 video traffic in ATM networks. Ahn et al. [1] propose a rate-based closed-loop congestion control scheme called Explicit Rate Indication Scheme for MPEG (ERISM) which is designed for MPEG-1 video transmission over Available Bit Rate (ABR) service. Priority-based mechanisms have also been proposed to minimize the effect of cell loss on the quality of the MPEG video [3, 23]. In [40], Mellaney et al. study the characteristics of MPEG-2 video traffic in an ATM network such as cell interarrival time and cell rate measurements. The use of MPEG-2 scalable video coding algorithms have been examined as an approach to cell loss resilience [5]. Cell packing methods have also been developed to conceal cell loss errors [17, 15].

Experiments investigating the relationship between MPEG video quality and

ATM network performance have also been conducted [62, 22]. In [62], Zamora et al. study the effect of cell errors, cell delay variation, AAL-5 PDU loss, and PDU size on the received video quality. They use a five-point impairment scale for subjective video quality assessment. In [22], Gringeri et al. discuss the effects of network impairments on video quality for MPEG-2 Transport Streams delivered over ATM. They also examine the issues of delivering VBR MPEG over ATM, such as the trade-off between bandwidth savings and video quality. However, they do not use quantitative video quality measurements in their studies.

With explosive growth of the Internet, intranets, and other IP-related technologies, it is our belief that a large majority of video applications in the near future will be implemented on personal computers using traditional best-effort IP-based networking technologies. Researchers have only recently begun considering the transport of MPEG video over IP networks. In [6], Basso et al. examine the use of the Real-Time Transport Protocol (RTP) defined by the Internet Engineering Task Force (IETF) to transport MPEG-2 streams over non-guaranteed quality of service (QoS) networks. In [48], Ramanujan et al. present an adaptive video streaming service for streaming MPEG video over a best-effort IP network environment. In [11], results from a study streaming MPEG-1 compressed video over the public Internet is presented. The study concentrates on network loss and error characteristics that specifically affect the quality of the received MPEG compressed streams. However, they use a frame error rate measure as a video quality metric rather than a more meaningful measure of received video quality such as the peak signal to noise ratio (PSNR) or the mean opinion score (MOS). In their experiments, they use low bandwidth and non-broadcast quality MPEG-1 video streams rather than higher quality MPEG-2 encoded streams.

In this thesis, we evaluate the effect of packet loss on the quality of MPEG-2 video transported over an IP network. In order to gain a better understanding of the characteristics of MPEG-2 video traffic in a best-effort IP network, an experimental video-on-demand (VoD) system has been employed. The system consists of UBC's Continuous Media File server (CMFS). Different load conditions for the network were considered. We studied the video quality degradation effects due to packet loss for three streams of differing motion activity levels. We used the objective PSNR quality metric and the subjective MOS quality metric to assess the video quality. We confirmed that the objective PSNR video quality metric is poorly correlated with subjective quality assessments. We observed that tiling, error blocks, motion jerkiness, and screen blanking were the most visible digital video distortions. We found that the activity level of the video influences the subjective quality assessments but not the objective quality measurements. The perceived video quality was found to be influenced by characteristics of the human visual system. We found that slice loss is linearly correlated with packet loss while picture header loss is poorly correlated with packet loss. We also found that slice loss is the dominant factor contributing to the degradation in video quality rather than picture loss. We found that low packet loss rates translate into much higher frame error rates due to the propagation of errors. Finally, we investigated the effectiveness of forward error correction, using Reed-Solomon coding, in protecting a video stream from packet loss. We found that FEC can increase a stream's packet loss rate tolerance significantly with only a small increase in overhead.

The rest of this thesis is organized as follows: First, a general overview of the MPEG-2 standard will be given in Chapter 2. Then a brief discussion of the networking requirements for audiovisual applications and a description of objective

and subjective video quality assessment techniques are given in Chapter 3. A survey of error control and concealment techniques for the transport of video over lossy networks is presented in Chapter 4. The experimental video-on-demand testbed network employed in our experiments and the data collection and analysis tools are described in Chapter 5. The various experiments along with the evaluation of the results for transporting constant bit rate MPEG-2 Elementary Streams over an IP network are presented in Chapter 6. Finally, Chapter 7 states our conclusions and proposes areas for future research.

Chapter 2

MPEG Overview

In order to monitor and measure MPEG video data streams over any type of network, it is important to first develop an understanding of the MPEG coding technology. In fact, it is important to know all the elements that make up an MPEG video stream and more importantly, to know their specific role in determining the quality of the video being transmitted.

In this chapter, we provide an overview of the MPEG-2 standard. We begin with a short history of MPEG standards, aspects of the human visual system, and then proceed to discuss the basic principles of MPEG coding.

2.1 History

In 1988, the ISO/IEC established the Moving Pictures Experts Group (MPEG) in order to define a standard for video and audio compression. MPEG's first effort led to the MPEG-1 standard, that was published in 1993 as ISO/IEC 11172. Today, it is mainly used in CD-ROM video applications such as CD-Interactive (CD-I) and Video-CD. It was designed to support video coding up to 1.5 Mbps with VHS

quality, audio coding at 192 kbps/channel (stereo CD-quality), and is optimized for non-interlaced video signals. MPEG-1 is not suited for broadcast environments or television applications, as it does not support all the features required for these applications.

In 1990, ISO/IEC started to work on the MPEG-2 standard. The main objective was to design a compression standard capable of different qualities depending on the bit rate, from TV broadcast to studio quality. The MPEG-2 standard is based on MPEG-1 but is more sophisticated and optimized for interlaced pictures. The MPEG-2 standard is capable of coding standard TV at about 4-9 Mbps and HDTV at 15-25 Mbps. The audio part of the standard adds multi-channel surround sound coding while being backwards compatible with the MPEG-1 Audio definition.

MPEG-2 is not the end of the story and ISO/IEC is moving forward to produce common standards for new applications of digital video. Currently, ISO/IEC is working on the MPEG-4 suite of standards. One of its objectives is to make it possible to mix and combine virtual images (like computer animations) and real video images on a bitstream level.

2.2 Aspects of the Human Visual System

Compression techniques, which are used in MPEG-2, are to a large extent based on the knowledge we have about how the eye and the visual centres in the brain recognize images.

The process of seeing involves two main tasks. First, the eye has to recognize details of a scene, which means it has to perceive the spatial resolution of the picture. The second task is to recognize changes in a scene, in other words, to perceive a temporal resolution of a scene.

The term *seeing* describes the idea that light reflected by the objects surrounding us enters our eyes. The eye itself contains several parts that process the reflected light and generate the image that our brain understands. When light has entered the eye, it passes through the cornea, the iris, the pupil, and finally the lens. All these parts work together to put a focused image onto the back of the eye, which is called the retina. Once the image is on the retina, it can be recognized and processed by the brain. The retina is equipped with photoreceptors which produce image information when stimulated.

There are two different kinds of photoreceptors: rods and cones. It was found that the rods allow us to see black and white while the cones allow us to distinguish between different colours. There are three different kinds of cones, which respond to incident radiation with somewhat different spectral response curves. If light is reflected on a high number of cones, the cones then enable us to get a high spatial resolution of the image since small changes in the colour can be recognized. Rods are more sensitive to the intensity of light itself and do not play a role in image reproduction.

An important aspect of the rods and cones in the context of digital video is their number and their distribution on the retina. Cones are distributed on the centre of the retina. Areas further away from the centre have a much higher distribution of rods. In total, we have 120 million rods and only around 8 million cones on the retina. The latter, as stated, are distributed close to the centre of the retina. This leads to the fact that the eye is, in general, relatively less sensitive to colour, especially to colour changes. Video compression techniques, like the one used in MPEG-2 Video, therefore utilize this low-colour sensitivity by reducing the colour information per image. MPEG-2 uses Discrete Cosin Transformation (DCT)

to indentify and subsequently remove high frequency changes in colour.

2.3 Coding Principles

To represent only one second of raw, broadcast quality (CCIR601), uncompressed video data, approximately 270 million bits are required. However, a video sequence typically contains spatial and temporal redundancies. These redundancies along with limitations of the human visual system are exploited in order to achieve high compression ratios. Spatial and temporal redundancies come from the fact that the pixel values are not completely independent but are correlated with the values of their neighbouring pixels both in space and time (i.e., in this frame and in subsequent or previous frames). This means that their values can be predicted to some extent. The human visual system has less acuity for higher spatial frequencies and is also less sensitive to detail near edges. Thus, the encoding process may be able to minimize the bit rate while maintaining constant quality of the picture to the human eye.

MPEG uses Discrete Cosine Transform (DCT) to deal with spatial redundancies (intraframe coding) and motion estimation for temporal redundancies (interframe coding). The specific quantization matrices that are used take into account the limitations of the human visual system.

2.4 Basic Units of MPEG-2 Video

A compressed video stream generated by an MPEG encoder has a hierarchical structure with different levels of components. This structure is shown in Figure 2.1.

Block: A block is the smallest coding unit in the MPEG algorithm. It is made up of 8x8 pixels and can be one of three types: luminance (Y), red chrominance

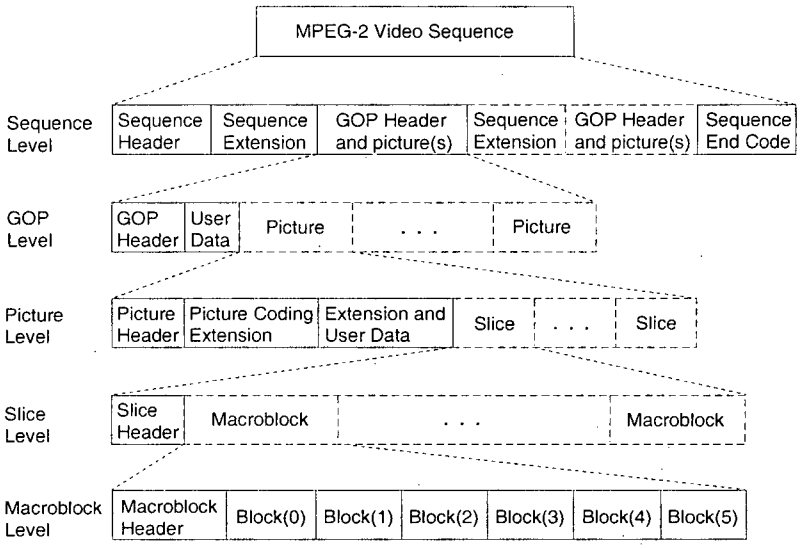


Figure 2.1: Structure of an MPEG-2 Video Elementary Stream

(C_r), or blue chrominance (C_b). The block is the basic unit in intraframe DCT coded frames.

Macroblock: A macroblock is the basic coding unit in the MPEG algorithm. It consists of a 16x16 pixel segment. Since MPEG's video Main Profile uses the 4:2:0 chroma format, a macroblock consists of four Y, one C_r, and one C_b block. It is the motion compensation unit.

Slice: A slice is a series of macroblocks and contains information about where to display the contained macroblocks on the screen. It is the basic element that allows support for random access within a picture. In the case of a transmission error and the loss of picture information, the information in a slice can be used to continue the display process within a picture. Instead of dropping the whole picture, the decoder can continue with the start of the next slice. Thus, slices serve as resynchronization units.

Picture: A picture in MPEG is a single frame of a video sequence.

Group-of-Pictures: A Group of Pictures (GOP) is simply a small sequence of pictures. It provides random access points into the video stream. The GOP concept was mandatory in the MPEG-1 standard whereas it is optional in the MPEG-2 standard.

Sequence: The sequence consists of a series of pictures (or a series of GOPs if GOPs are present).

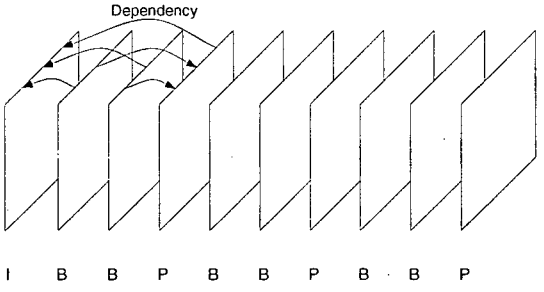
2.5 Picture Types

In MPEG-2, there are three types of pictures:

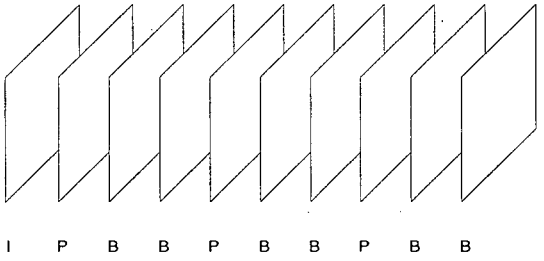
Intra-coded or I-pictures: These pictures are coded indepedently, entirely without reference to other pictures. Temporal redundancy is not taken into account. Moderate compression is achieved by exploiting spatial redundancy. An I-picture is always an access point into the video stream.

Predictive-Coded or P-pictures: These pictures are coded with respect to a previous I- or P-picture. The information that can be used from the previous picture is determined by motion estimation. Information that cannot be 'borrowed' from previous pictures is coded in the same way I-pictures are coded. Thus, the coding process here exploits both spatial and temporal redundancies. P-pictures are around 30 - 50% of the size of an I-picture.

Bidirectionally-Predicted or B-pictures: These pictures use both previous and future I- or P-pictures as a reference for motion estimation. They achieve the highest compression ratios, approximately 50% of the size of a P-picture.



(a)



(b)

Figure 2.2: MPEG Video (a) Display Sequence and (b) Coding Sequence

Since they reference both past and future pictures the coder has to reorder the pictures such that the coding order, the order in which compressed pictures are found in the bitstream, is not the same as the display order, the order in which pictures are presented to the viewer. Figure 2.2(a) shows a typical MPEG video display sequence while (b) shows the coding (and transmission) order.

2.6 Data Compression Used in MPEG-2 Video

The basic MPEG algorithm consists of the following stages: a motion estimation stage, a transformation stage, a lossy quanitzation stage, and a lossless coding stage.

The motion estimation stage takes the difference between the current image and the adjacent image. The transformation stage then tries to concentrate the information energy into the first transform coefficients. The quantization step that follows causes a loss of information, and the final stage is an entropy encoding stage that further compresses the data.

MPEG is a lossy compression scheme since the reconstructed picture is not identical to the original. High compression ratios cannot be achieved without using lossy compression techniques because the least significant bits of each colour component become progressively more random and, therefore, harder to code.

2.6.1 Motion Estimation

Motion estimation is used for interframe prediction in order to exploit temporal redundancies found in the video sequence. It is used in both B- and P-pictures. The idea behind motion estimation is to identify regions in the picture that can be found in the following picture as well. Since the pictures occur at rates of 20-30 per second, it is very likely that similar, but maybe slightly moved, regions can be detected in adjacent pictures. The motion estimation process use the macroblocks as basic units for comparison. For each macroblock, the encoder searches the previous (in the case of a P-picture) or the previous and the future picture (in the case of a B-picture) for a macroblock that matches or closely matches the current macroblock. If such a macroblock is found, the difference between this macroblock and the current macroblock is calculated. The resulting difference is first DCT coded, quantized, and then, together with the motion vector of the macroblock, entropy coded.

2.6.2 DCT Coding

An image is divided into blocks and Discrete Consine Transform (DCT) is used to approximate the original chrominance and luminance information of each block. Instead of using the real colour values for each block, a set of frequency coefficients is calculated which describes the colour transitions in the block. However, the DCT does not reduce the number of bits required from the block representation. The reduction is being done after the observation that the distribution of coefficients is non-uniform. The transformation concentrates as much of the video energy as possible into the low frequencies leading to many coefficients being zero or almost zero. The compression is achieved by skipping all those near zero coefficients and by quantizing and variable-length coding the remaining ones.

2.6.3 Quantization

Quantization reduces the number of possible values of DCT coefficients and, thereby, reducing the required number of bits. The quantization stage takes advantage of the spatial frequency dependency of the human eye's response to luminance and chrominance [46]. It has been shown that numerical precision of the DCT coefficients may be reduced without affecting image quality significantly. Thus, each coefficient is weighted according to its impact on the human eye (i.e., high-frequency coefficients should be more coarsely quantized than low-frequency ones).

MPEG-2 defines default quantization matrices, but also allows user-defined quantization matrices. Quantization is also controlled by a scale factor, which allows the user to adjust the quantization level (and by this, also the compression ratio). By having a scaling factor in the quantization process, it becomes possible to generate constant bit rate video streams, which fit into the constraints that might be given

by a certain network architecture.

2.6.4 Entropy Coding

After the quantization stage, the quantized DCT coefficients are serialized and arranged in a zig-zag scanning order, as shown in Figure 2.3, enabling further compression through entropy (lossless) coding. The scan approximately orders the coefficients in ascending spatial frequency. The DCT and quantization stages have concentrated much of the video energy into the low frequencies and as a result, a large number of zeros in the high frequency range. This is important because it becomes easy to code the resulting row of numbers efficiently using entropy coding techniques. In MPEG, variable length coding (VLC) is used to entropy-code the sequence of coefficients. It uses very short codes (only a few bits) for patterns that occur very often in the sequence.

2.6.5 Constant Bit Rate vs. Variable Bit Rate

MPEG-2 video streams can be encoded in one of two ways: constant bit rate (CBR) or variable bit rate (VBR). In general, the bit rate of compressed video streams are variable bit rate and vary according to the content of the video sequence. If a VBR stream is to be transmitted over a fixed rate channel, the variations must be smoothed out to produce a constant bit rate. CBR streams are buffer regulated to allow continuous transfer of coded data across a fixed rate channel without causing an overflow or underflow of the buffer. The encoder accomplishes this by monitoring the buffer's occupancy and adjusting the coding parameters, such as the quantization parameter Q, so the bit rate can be kept constant throughout the stream. The Q parameter defines the step size for the scaling of the quantization; the lower the

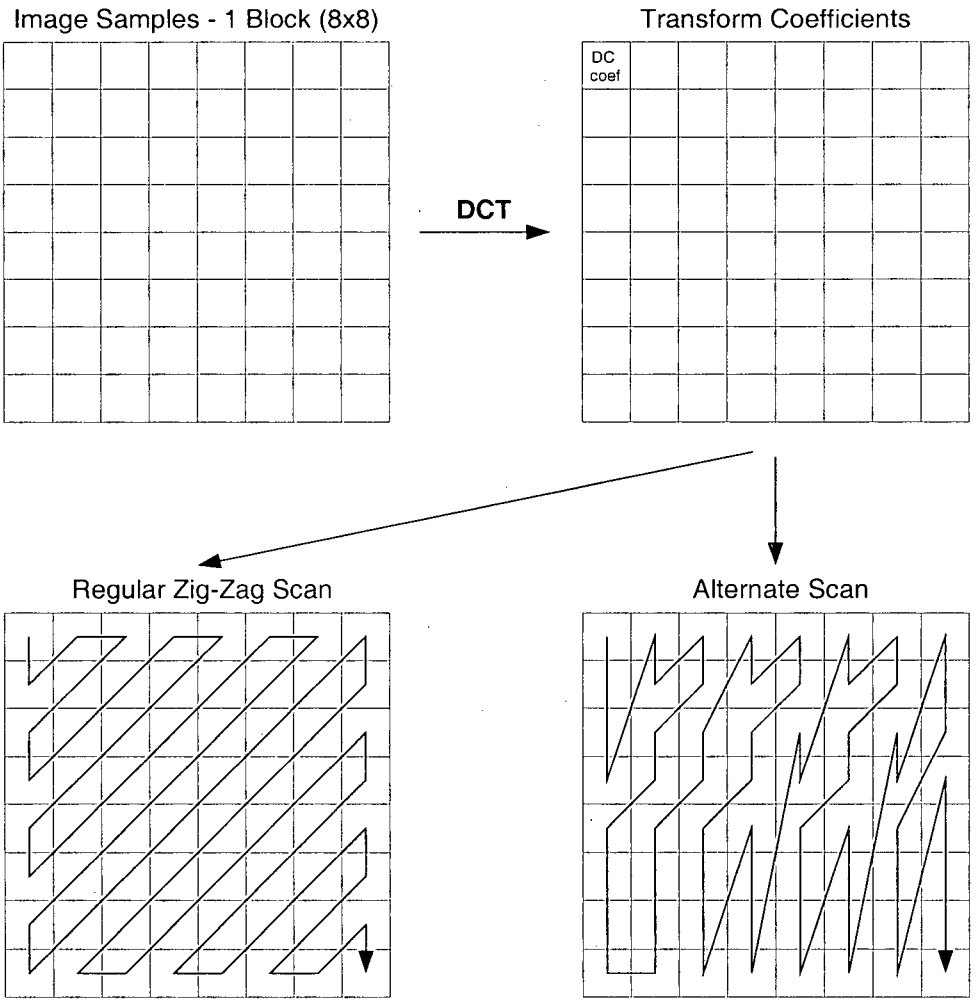


Figure 2.3: Zig-Zag Scanning Order of Coefficients

value of Q, the better the quantitative quality of the encoded video.

The main disadvantage of CBR video coding is that maintaining a constant bit rate is usually at the expense of variable picture quality. In contrast, VBR coding does not have any buffer constraints allowing it to maintain constant picture quality. Constant picture quality is achieved by holding the quantization step size Q constant for all frames in the sequence. Since the content of a video sequence changes from frame to frame, the number of bits generated for each frame will also vary.

Chapter 3

Video Quality

In this chapter, the audiovisual requirements from a networking point of view are examined. Although the focus of this research is on video, the network requirements for audio services is provided for comparison purposes. We then describe various video quality assessment techniques. Finally, the perceptual impact of losses is presented.

3.1 Network Requirements

The network requirements for audiovisual applications depend primarily on the application scenario. There are three main application scenarios: interactive audio, interactive video, and non-interactive video. These application scenarios can be distinguished according to the end-to-end delay and bandwidth requirements. Table 3.1 summarizes the delay and bit rate requirements for the three application scenarios.

Depending on whether the application is interactive or not, the delay requirements differ. For audio conferencing, there is little impact if the delay is less than

Application Scenario	Delay	Bit Rate
Interactive Audio	< 200 ms	8 - 64 Kbps
Interactive Video	200 - 500 ms	64 Kbps - 2 Mbps
Non-Interactive Video	1 - 5 sec	2 - 8 Mbps

Table 3.1: Application Scenario Delay and Bit Rate Requirements

150 ms, but serious degradation in quality if the delay is more than 400 ms [31]. For video conferencing, synchronization between the video and the associated audio stream requires tight delay bounds. It has been shown that video can precede the associated audio by up to 100 ms or follow it by at most 20 ms [12]. Non-interactive applications, such as video-on-demand, can tolerate much higher delays on the order of a second or more, since their delay is only noticeable as startup delay, but transparent after playout starts.

Bit rate requirements vary according to the encoding scheme used as well as the actual content. Pulse code modulation (PCM) is the standard codec used for interactive audio applications and has a data rate of 64 Kbps. Other codecs such as adaptive differential PCM (APCM), G.723.1, and G.729 have been developed to provide low bit-rate internet telephony services that operate in bandwidth-, delay-, loss-, and cost-constrained environments [33]. Interactive video applications typically use the H.261 codec which has a data rate between 64 Kbps - 2 Mbps. Non-interactive video applications typically use the MPEG-2 codec that is capable of providing VCR quality (2 Mbps) to broadcast quality video (8 Mbps) data rates. For MPEG-2 VBR video, the data rate will vary depending on the amount of scene motion. High scene motion will prevent high temporal compression which results in an increase of the data rate.

3.2 Video Quality Assessment

The methods of video quality assessment can be divided into two main categories: subjective assessment and objective assessment. Subjective measurements are the result of human observers providing their opinion of the video quality. Objective measurements are performed with the aid of instrumentation that incorporates mathematical algorithms.

3.2.1 Subjective Measurement

Television programs and other video material are produced for the enjoyment or education of human viewers so it is their opinion of the video quality which is most important. Informal and formal subjective measurements have always been, and will continue to be used to evaluate video quality. Even with all the objective testing methods available today for analog and digital video, it is important to have human observation and assessment of the video because there are impairments which are not easily measured yet are obvious to a human observer. Therefore, casual or informal subjective testing remains an important part of system evaluation.

Formal subjective testing, as defined by ITU-R BT.500 [30], has been used for many years with a relatively stable set of standard methods until the advent of digital compression. With digital compression, the picture quality is not constant over time. Picture quality is a function of complexity of the program material and, in the case of statistical multiplexing, the moment to moment operation of the transmission system. Considering this time varying property and the number of new impairments, the defined and proposed measurement methods have grown in recent years. In brief, formal subjective testing involves selecting a number of non-expert observers, testing them for their visual capabilities, showing them a series of test

Scale	Impairment	Quality
5	Imperceptible	Excellent
4	Perceptible, but not annoying	Good
3	Slightly Annoying	Fair
2	Annoying	Poor
1	Very Annoying	Bad

Table 3.2: Subjective Quality Scale

scenes for about 10 to 30 minutes in a controlled environment, and asking them to score the quality of the scenes in one of a variety of manners. One of the grading scales defined in ITU-R BT.500 is a five-point quality and impairment scale, shown in Table 3.2. A mean opinion score (MOS) is usually generated by averaging the viewer ratings.

The advantages of subjective testing are that it produces valid results for both conventional and digitally compressed video systems, a scalar mean opinion score is obtained, and it works well over a wide range of still and motion picture applications. The weakness of subjective testing is that it is very complex which makes it extremely time consuming. Subjective testing does not lend itself to operational monitoring, production line testing, or trouble shooting.

3.2.2 Objective Measurement

The need for an objective testing method of picture quality is clear, subjective testing is very complex and provides too much variability in results. It is useful to have objective testing methods which are repeatable, can be standardized, and can be performed quickly and easily. However, since it is the human observer's opinion of picture quality that counts, any objective measurement system must have good correlation with subjective assessments.

A wide variety of methods have been investigated for objective testing of

picture quality in compressed video systems. The methods can be roughly divided into two categories: feature extraction and picture differencing.

Feature extraction uses a mathematical computation to derive characteristics of a single picture (spatial features) or a sequence of pictures (temporal features). The calculated characteristics of the reference and degraded pictures are then compared to determine an objective quality score. A recently approved American National Standards Institute (ANSI) standard [4] defines several feature extraction measurements such as maximum added edge energy, maximum lost motion energy, and average motion energy difference. However, research at Tektronix and other laboratories have shown that certain picture differencing methods provide better objective picture quality measurement correlation with subjective results [18].

Picture differencing uses a matrix-based mathematical computation to process each picture or sequence of pictures. Usually, the pixel-by-pixel differences, after synchronization, of the undecoded (reference) pictures and the coded pictures are used to compute comparative distortion measure [45] such as the following:

- Mean Square Error (MSE),

$$MSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (x_{i,j} - y_{i,j})^2}{NM}$$

- Peak Signal to Noise Ratio (PSNR),

$$PSNR(dB) = -10log\left(\frac{MSE}{(x_{i,j})^2_{max}}\right)$$

However, these measurements are only effective when the errors are an additional noise and not correlated to the signal content. They cannot distinguish between few errors with high amplitude, which are annoying for final viewers, and

many impairments with lower amplitude, which may be subjectively imperceptible. Unfortunately, the correlation of these objective measurements with subjective measurements is commonly known to be quite low [45, 16].

In recent years, researchers have been developing perceptual objective models [60, 34, 38, 58] that are based on an approach that takes into account some of the known properties of the human visual system (HVS), and in this way perform objective quality measurements that predict the results of subjective measurements. A spatio-temporal model of human vision has been developed in [34, 35] for the assessment of video coding quality. A new metric is introduced based on a multi-channel model of human spatio-temporal vision that has been parameterized for video coding applications by psychological experiments. In [38], the Just Noticeable Difference (JND) model, a physiologically-based model of human visual discrimination performance, is introduced to automatically and accurately assess the perceptual magnitude of difference between a test and reference sequence.

Due to the complexity and unavailability of perceptual-based objective measurement algorithms, we have chosen to use the PSNR measurement that has been used extensively in other literature along with an informal subjective measurement since subjective assessment still remains the most reliable instrument to evaluate picture quality.

In our project, we are interested in the effects of network transmission impairments on the quality of coded pictures. We take the decoded pictures of an MPEG-2 stream as the reference pictures and compare them to the decoded pictures of the transmitted MPEG-2 stream. Therefore, the video quality measurements will only take into account video impairments that are due to network transmission errors rather than codec errors.

3.3 Perceptual Impact of Losses

In a networking environment, the factors that have the greatest impact on quality are the actual packet delays, the number of lost packets, the number of pixels in an impaired region and its shape, and the burstiness of the losses [31]. The extent of picture degradation is strongly dependent on the type of video information lost.

Video as well as audio services have a real-time aspect and are delay sensitive. Data arriving beyond a certain point in time is considered lost or meaningless by the application.

In [31], random packet losses were found to yield greater quality degradation than clustered losses at equal loss ratios. Thus, for a given loss probability, it may be safe to assume that uncorrelated loss events give an upper bound on the quality degradation.

It is clear that data loss reduces the quality of the transmitted picture. The extent of the degradation is strongly dependent on the type of information lost. Losses of headers and system information affect the quality differently than losses in pure video data. Furthermore, the quality reduction depends also on the location of the lost video data due to the predictive structure of an MPEG encoded video stream. Data loss spreads within a single picture up to the next resynchronization point (slice headers) mainly due to the use of variable length coding, run length, and differential coding. This is referred to as spatial propagation and may damage any type of picture. When loss occurs in a reference picture (intra-coded I-pictures or predictively coded P-pictures), the error will remain until the next I-picture is received. This causes the error to propagate across several non-intra-coded pictures until the end of the group-of-pictures. This is known as temporal propagation and is due to inter-frame prediction. Errors in B-pictures do not propagate because

B-pictures are not used in the prediction of other pictures.

Chapter 4

Error Control and Concealment

One inherent problem with any communication system is that information may be altered or lost during transmission due to either channel noise or congestion. The effect of information loss can be devastating for the transport of MPEG-2 video because any damage to the compressed data stream may lead to objectionable visual distortion at the receiver. Therefore, error control and concealment are necessary for minimizing the impact of data loss.

Transmission errors can be roughly classified into two categories: bit errors and data loss. Bit errors are caused by imperfections in the physical channel which can result in bit inversions. Data loss is caused by packet loss which is usually a result of packets being dropped at congested routers and switches. In the case of UDP, the integrity of the data payload is validated using a CRC. If a bit error is detected, the packet is discarded and never passed up to a higher layer. Therefore, bit errors are treated as data loss errors.

Techniques for combating transmission errors for video communication have been developed along two avenues: lossless and lossy recovery. Traditional error con-

trol and recovery schemes for data communications have been extended for video transmission which aim at lossless recovery. Examples of such techniques include forward error correction (FEC) and automatic retransmission request (ARQ). Development of lossy recovery techniques involving signal-reconstruction that strive to obtain a close approximation of the original signal or attempt to make the output signal at the decoder least objectionable to the human eye. Lossy recovery techniques are possible because unlike data transmission, where lossless delivery is required absolutely, human eyes can tolerate a certain degree of distortion in image and video signals.

4.1 Lossless Recovery

Forward error correction (FEC) is an error concealment technique in which repair data is added to a media stream, such that packet loss can be repaired by the receiver of the stream without further reference from the sender. Basic FEC schemes involve exclusive-OR operations, the idea being to send every kth packet a redundant packet obtained by exclusive-ORing the other k packets [54]. Such mechanisms can recover from a single loss in a k packet message. However, they add latency since k packets have to be received before the lost packet can be reconstructed.

The potentional of FEC schemes to recover from losses depends on the characteristics of the packet loss process in the network. FEC mechanisms are more effective when lost packets are dispersed throughout the stream of packets. If losses are very bursty, then basic FEC mechanisms are not as effective. Measurements of packet loss characteristics in the Internet have shown that the vast majority of losses are of single packets [7, 8]. Burst losses of two or more packets are around an order of magnitude less frequent than single packet loss. Longer burst losses (of the

order of tens of packets) occur infrequently but account for most of the loss rate [10]. Other FEC schemes that are more sophisticated and computationally demanding, such as Reed-Solomon (RS) coding, are usually structured so that they have better burst loss protection.

Retransmission of lost packets is another obvious lossless recovery technique by which loss may be repaired. Retransmission has a high latency since a retransmission request must first be sent by the client to the server before the lost packet is retransmitted. Therefore, it is not suitable for interactive audio and video applications that have tight delay bounds. Measurements have shown that in large Mbone sessions, most packets are lost by at least one receiver [24]. The high overhead of requesting retransmission for most packets make it unsuitable for multicast applications. In this case, the use of forward error correction may be more acceptable. A combination of FEC and retransmission as an error recovery technique is discussed in [44]. FEC is used to repair all single packet losses, and those receivers experiencing burst losses, and willing to accept the additional latency, use retransmission as an additional recovery mechanism.

Interleaving is another technique that can be used to reduce the effects of packet loss. When the unit size of the data is smaller than the packet size and the receiver can tolerate higher latency, interleaved packetization can be used where successive units are put into nonadjacent packets. This prevents the loss of contiguous blocks and disperses the effect of packet losses. The major advantage of interleaving is that it does not increase the bandwidth requirements of a stream. For MPEG video ES streams, interleaving is not effective unless the packet sizes are large to allow multiple MPEG slices to fit into one packet.

Another preprocessing error concealment technique is scalable or layered

video coding [9, 5] which prioritizes the data. The encoded data is segmented into low priority data such as high frequency DCT coefficients, and high priority data such as addresses of blocks, motion vectors, and low frequency DCT coefficients. Such prioritization is performed since motion vectors are crucial to the reconstruction of motion predicted regions. Likewise, most of the information for reconstructing blocks of data are available in the low frequency DCT coefficients. The idea is to transmit the high priority data through a high priority, lossless channel and the low priority data is sent over a low priority, lossy channel. In the event of packet loss in the low priority data stream, the high priority layer data can still be used to reassemble a viewable video sequence.

The MPEG-2 standard provides four methods to produce a layered bitstream: data partitioning, signal-to-noise (SNR) scalability, spatial scalability, and temporal scalability. All four generate at least two bitstreams: the base-layer and the enhancement-layer bitstreams. The base-layer bitstream can be decoded independently to generate a lower quality version of the encoded video. The main difference between the various techniques is in the content of the base layer. In the data partitioning technique, the base layer contains a reduced set of DCT coefficients. In SNR scalability, the base layer consists of a coarsely quantized version of the video. In spatial scalability, the spatial resolution of the base layer is reduced. Last, in temporal scalability, the base layer has a reduced temporal resolution. In all cases, the enhancement layer contains the necessary information to obtain the high-quality video.

The combination of the base layer and the enhancement layer is dependant upon the layered coding method. Data partitioning multiplexes the two sets of DCT coefficients back together to obtain the original non-layered bitstream before

being decoded. If losses have occurred, the loss concealment logic generates an appropriate 'filler code' to create a syntactically correct non-layered bitstream. The simplest filler code is one that corresponds to zeros for the lost coefficients. For SNR scalability, the base- and enhancement-layers are independently processed to obtain the dequantized DCT coefficients. The two sets of coefficients are then summed blockwise before the inverse DCT is applied to this sum. The reader is referred to [5, 29] for further details on spatial scalability and temporal scalability decoding.

The MPEG-2 standard allows for hybrid scalability, the combination of two different types of scalability. The types of scalability that can be combined are SNR scalability, spatial scalability, and temporal scalability. When two types of scalability are combined, there are three bitstreams that have to be decoded: a base layer and two enhancement layers. Hybrid scalability allows more demanding applications to be supported. For example, spatial and SNR hybrid scalability enables support for standard TV and HDTV at two qualities with the base layer providing standard TV resolution. Using spatial scalability, enhancement layer 1 provides basic HDTV quality and enhancement layer 2 uses the SNR scalability to help generate high quality HDTV.

4.2 Lossy Recovery

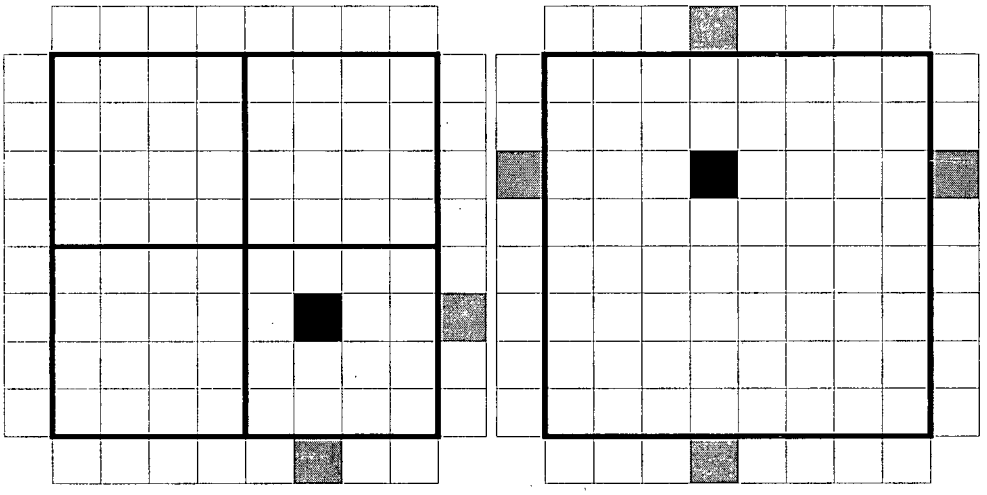
Error concealment techniques for MPEG-2 video communication based on lossy recovery are possible due to the characteristics of the human visual system. It is well known that images of natural scenes have predominantly low-frequency components, i.e., the colour values of spatial and temporally adjacent pixels vary smoothly, except in regions with sharp edges. In addition, the human eye can tolerate more distortion to the high-frequency components than to the low-frequency components.

These facts can be used to conceal the video artifacts caused by transmission errors. Lossy recovery schemes perform postprocessing at the decoder and make use of the redundancy present in the image in the frequency, spatial, and temporal domains [14]. Under these schemes part of the lost information can be recovered by regenerating most of it by means of interpolation in each domain.

The lossy recovery error concealment schemes work at the macroblock (consisting of several blocks) level of the MPEG-2 video hierarchy. The MPEG-2 slice unit is a synchronization point that begins each scan row of macroblocks and, therefore, a transmission error will only cause damage to a single row, so that the upper and lower macroblocks of a damaged block may still be correctly received. Also, the transmission error events among two adjacent frames are usually sufficiently uncorrelated so that for a given damaged macroblock in the current frame, its corresponding macroblock in the previous frame is usually received undamaged [59]. All the lossy recovery schemes make use of the correlation between a damaged macroblock and its adjacent macroblocks in the same frame and/or the previous frame to accomplish error concealment.

One simple way to exploit the temporal correlation in a video signal is by replacing a damaged macroblock with the spatially corresponding macroblock in the previous frame. However, this method can produce adverse visual artifacts in the presence of excessive motion or scene changes. More sophisticated mechanisms are required in the event of excessive image motion or loss.

The MPEG-2 video coding process employs the Discrete Cosine Transform (DCT) to reduce the spatial redundancy found on the original video sequence. DCT transforms a data block in the spatial domain into a block of the same size in the spatial frequency domain. Error concealment schemes operating in the frequency



(a) (b)

Figure 4.1: Spatial Interpolation using (a) 2 nearest boundaries, (b) all 4 boundaries

domain relies on the fact that the low frequency DCT coefficients contain most of the background information and that there is high correlation between adjacent blocks. These schemes perform a linear or polynomial interpolation of adjacent DCT coefficients [25, 55].

Error concealment schemes operating in the spatial domain perform interpolation of each pixel in the damaged block from pixels in the adjacent blocks [21, 2] in the same frame. In [2], two methods are proposed to interpolate pixel values. In the first method, a pixel is interpolated from two pixels in its two nearest boundaries, as shown in Figure 4.1(a). In the second method, shown in Figure 4.1(b), a pixel in the macroblock is interpolated from the pixels in all four boundaries.

The major advantage of spatial error concealment schemes is that they are applicable to I-pictures and scene changes. The major drawback of these schemes is the high computational cost.

The MPEG-2 compression algorithm employs motion estimation to generate motion vectors for macroblocks. Temporal error concealment schemes make use of the motion vectors to recreate a missing block. Based on the same assumption about spatial smoothness, lost or damaged motion vectors can be similarly interpolated from that of spatially or temporally adjacent blocks. For estimation of lost motion vectors, the following methods have been proposed [59]: 1) setting the motion vectors to zero; 2) using the motion vectors of the corresponding block in the previous frame; 3) using the average of the motion vectors from spatially adjacent blocks; 4) using the median of motion vectors from the spatially adjacent blocks.

The first method works well for video sequences with relatively small motion. It has been found that the last method, as shown in Figure 4.2, produces the best reconstruction results [41]. The scheme works well under the assumption that adjacent macroblocks move in the same direction to that of the lost block. However, not all the motion vectors of all the adjacent macroblocks are always available as some of them may have been lost or coded as intra blocks, in which case they do not have an associated motion vector, preventing accurate estimation. Temporal error concealment schemes offer good results for most sequences except those presenting high and irregular motion levels and scene changes. In the latter case, spatial error concealment schemes provide better results [14]. Temporal error concealment schemes also cannot be used for I-picture reconstruction since there are no associated motion vectors for the macroblocks for these intra-coded pictures.

4.3 Reed-Solomon Coding

FEC has been proposed to compensate for packet loss for applications such as multimedia that have strict delay requirements that eliminate the possibility for re-

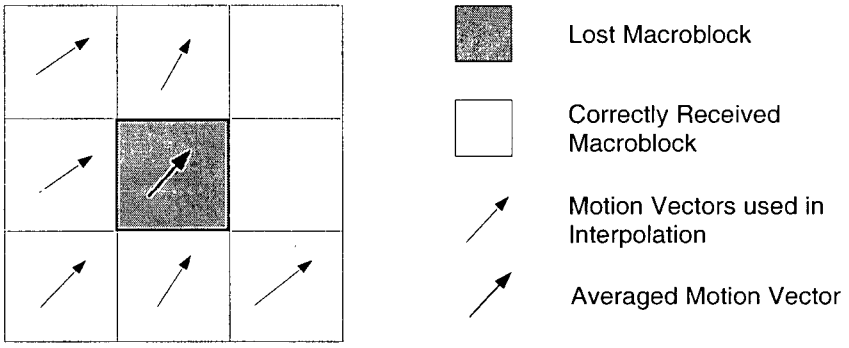


Figure 4.2: Motion Vector Estimation using Median Vector

transmissions. It has also been proposed for use in multicast communication where retransmission requests by receivers does not scale well since losses are uncorrelated between receivers [50]. In such cases, FEC techniques can be used to allow the receivers to recover from independent losses. In our project, we have employed FEC using Reed-Solomon coding.

Reed-Solomon codes are block-based error correcting codes. The key idea behind RS coding is that the encoder takes a block of data and adds extra "redundant" bits. Errors occur during transmission which corrupt the coded block. The decoder processes each block and attempts to correct errors and recover the original data. The number and type of errors that can be corrected depends on the characteristics of the Reed-Solmon code employed.

A Reed-Solomon code is specified as RS(n,k) with s-bit symbols. This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are n-k parity symbols of s bits each. Given a symbol size of s, the maximum codeword length is $n = 2^s - 1$.

A RS decoder can correct up to t symbols that contain errors in a codeword, where $2t = n - k$. One symbol error occurs when 1 bit in a symbol is wrong or when all the bits in a symbol are wrong. For example, RS(255,223) with 8-bit symbols

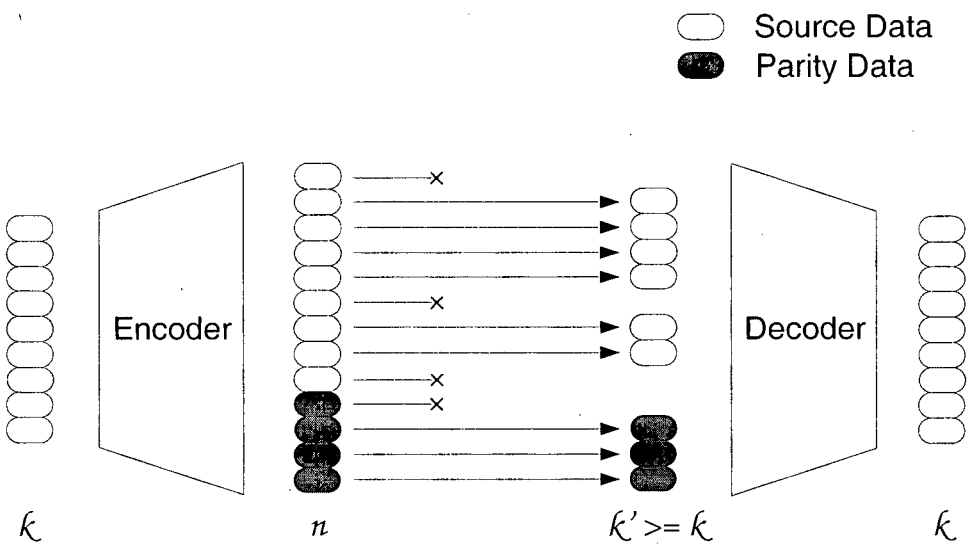


Figure 4.3: A graphical representation of the Reed-Solomon Encoding/Decoding process

can correct 16 symbol errors. In the worst case, 16 bit errors may occur, each in a separate symbol (byte) so that the decoder corrects 16 bit errors. In the best case, 16 complete byte errors occur so that the decoder corrects 16 x 8 bit errors. Thus, a decoder can correct up to t errors or $2t$ erasures. Under erasure-only conditions, a RS decoder can recover the source data given any subset of k symbols (see Figure 4.3).

Reed-Solomon codes may be shortened by (conceptually) making a number of data symbols zero at the encoder, not transmitting them, and then re-inserting them at the decoder. For example, a RS(255,223) code can be shortened to RS(200,168). The coder takes a block of 168 data bytes, (conceptually) adds 55 zero bytes, creates a (255,223) codeword and transmitts only the 168 data bytes and 32 parity bytes. At the decoder, the 55 zero bytes are added back in before the block is decoded.

The theory behind Reed-Solomon coding is beyond the scope of this paper. The reader is referred to [50, 61] for further details.

Chapter 5

Experimental System

In order to gain a better understanding of the characteristics of MPEG-2 video traffic, a video-on-demand (VoD) system has been setup to conduct experimental studies. The system consists of a video server and a video client implemented on personal computers which are connected together via Fast Ethernet LANs.

5.1 Video Server and Client

The video-on-demand testbed system has been implemented following a client-server paradigm. The video server used in the VoD system is UBC's Continuous Media File Server (CMFS) [42]. It was designed to support the storage and retrieval of time-sensitive continuous media such as video and audio.

Continuous media streams are stored in the server as media objects. A media object is broken down into a sequence of data objects called *segments*. The segment boundaries are determined at the time the media object is created and may depend on the syntax of the media. In this case, an MPEG-2 video elementary stream is stored as a sequence of group-of-picture (GOP) segments. Each media object is

assigned a universally unique object identifier (UOI) which is used to locate the object when a client requests for it.

The CMFS server was designed as a distributed service consisting of an administrator node and a set of server nodes, each with a processor and disk storage. However, for the purposes of this project, our interest is not in the scalability of the server and, thus, we have run the administrator node and a single server node as separate processes on a single PC workstation. In particular, the CMFS server runs on an HP Vectra 200 Mhz Pentium Pro PC running FreeBSD 3.0, with a 2 GB SCSI-2 Fast/Ultra hard drive.

CMFS uses a custom transport protocol called Media Transport [39]. It is a simple unreliable stream protocol which uses an underlying datagram protocol, such as UDP, for transmission of data. An MT connection is neither flow-controlled nor error controlled. CMFS's application-level flow control scheme is used in place of a transport-level flow control scheme. The MT protocol does not include an error control scheme because it would imply retransmission which is considered inappropriate for real-time data streams. A separate reliable TCP connection is made between the client and server for passing critical control messages.

CMFS divides time into small units called *slots*. Each slot is 500 milliseconds. Data is retrieved from the disk and sent across the network in units of slots. For example, 15 video frames of a 30 frame per second video stream may be the amount of data retrieved and sent during a slot.

A client requests the presentation of a media object from the server using three RPC requests: *CmfsOpen*, *CmfsPrepare*, and *CmfsRead*. A CmfsOpen request, which includes the UOI of the requested object, is sent to the server administrator which determines if the object exists. During open, the attributes of the object

are retrieved by the server node to determine the data rate and client buffering requirements for the object. The minimum amount of client buffering is the data required in the largest two consecutive slots. This is because the model requires double buffering: both the consumption of bits as they are displayed and the arrival of bits from the server proceed at variable rates. Based on these calculations, the server requests a real-time MT connection to be opened from the server to the client. The client completes the connection establishment by setting the values for buffer space that it is willing to devote to the stream. Control from the CmfsOpen request is returned back to the client with indication of success if the connection parameters are acceptable to the server as well. A conection identifier is issued and used to identify the real-time connection in all control requests.

When a client is ready to begin receiving the data, it sends a CmfsPrepare request to the server. The server performs an admission test of the disk and network requirements of the media stream. If successful, the server schedules all disk reads required for the duration of the stream. When the first slot of data has been read from the disk and sent to the client, the CmfsPrepare request returns to the client. This prevents the client from underflowing as well as to help minimize delay variation effects. Once the stream is prepared, the client can begin reading and processing the data via CmfsRead requests.

5.2 End-to-End Flow Control

CMFS utilizes a credit-based flow control protocol that is controlled by a Network Manager thread running at the server node. The server node also consists of Stream Manager threads, one for every active stream and a Disk Manager thread. A stream manager is responsible for the actual network transmission of data. The disk man-

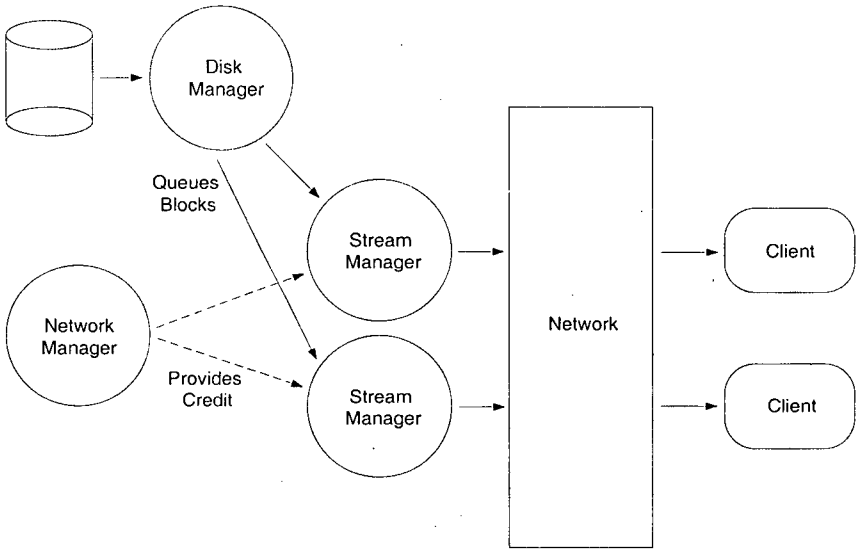


Figure 5.1: Structure of Server Node

ager is responsible for reading disk blocks according to its schedule and enqueues them for the appropriate stream manager for transmission. Figure 5.1 shows the structure of the server node.

The flow control of a stream is handled using credits which are issued to the stream managers, by the network manager, before they can send data across the network. Without flow control of some kind, the stream managers would send as fast as the network would allow or as fast as the disk could read, causing overflow at the server, network switch, or client. The network manager has knowledge of the rate of each connection and the amount of buffer space at each client as well as the amount of data to be displayed per slot. This information allows the network manager to issue the appropriate amount of credit to prevent overflow and underflow.

The flow control protocol utilizes a start packet which is sent to the server on the first client CmfsRead request to notify it that the client has begun to read. No further communication from the client to the server is necessary, because the

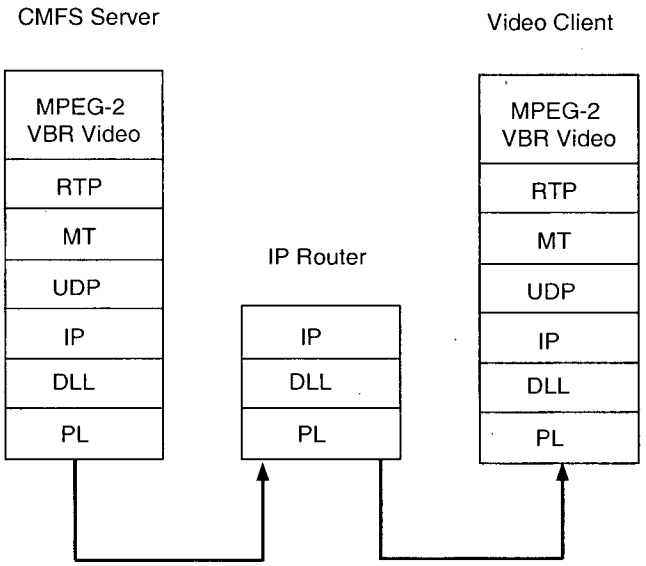


Figure 5.2: Protocol Architecture of the VoD System

server then assumes that the client will continue to consume data at the rate which was specified in the CmfsPrepare request. At the beginning of each time slot, the network manager decreases its record of available client buffer space by the amount of data required to be sent for the current slot. The network manager also issues credit to the stream managers for the current slot if data must be sent at this time. Further details regarding the flow control protocol can be found in [43].

5.3 Network Protocol Architecture

The Real-Time Transport Protocol (RTP) [52] is used to encapsulate the MPEG video data in our experimental VoD system. The RTP protocol data units (PDUs) are encapsulated in CMFS's Media Transport PDUs. The complete protocol architecture of our experimental VoD system is given in Figure 5.2.

5.3.1 Real-Time Transfer Protocol (RTP)

The growth of the Internet and the demand for multimedia applications have led to the development of the Real-Time Transfer Protocol (RTP) [52] by the Internet Engineering Task Force (IETF). RTP is designed to deliver various kinds of real-time data over packet networks that include non-guaranteed quality-of-service networks. RTP addresses the needs of real-time data transmission only and relies on other well-established network protocols for other communication services such as routing and multiplexing.

The services provided by RTP include payload type identification, sequence numbering, time stamping, and delivery monitoring. RTP typically runs on top of User Datagram Protocol (UDP) to make use of its multiplexing and data checksum services. This is in addition to the basic networking services provided by the underlying IP layer. However, RTP may be used with other suitable network or transport protocols. RTP also supports data transfer to multiple destinations using multicast distribution if provided by the underlying network.

RTP does not provide any mechanism to ensure timely delivery or provide other QoS guarantees. This requires the support of lower layers that actually have control over resources in switches and routers. RTP does not guarantee delivery or prevent out-of-order delivery. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence as well as detect lost packets.

The payload type identification service of RTP together with the multiplexing services supported by the underlying transport protocol, such as UDP, provides the necessary infrastructure to multiplex a large variety of information effectively. The transmission of an MPEG audio and video stream multiplexed together with any other auxiliary information can easily be handled using these services. The time

stamping service provides for encoder-decoder clock matching as well as synchronization of multiple sources.

RTP is based on the Application Level Framing (ALF) and Integrated Layer Processing (ILP) [13] principles, which dictate using the properties of the payload in designing a data transmission system as much as possible. For example, if we know that the payload is MPEG video, we should design our packetization scheme based on 'slices' because they are the smallest independently decodable data units for MPEG video. This approach provides a much more suitable framework for MPEG-2 transmission over networks with high packet loss rates. RTP provides basic packet format definitions to support real-time communication but does not define control mechanisms or algorithms. RTP is intended to be adaptable and tailored through modification and/or addition to the headers to provide the information required by a particular application and will often be integrated into the application processing rather than implemented as a separate layer.

RTP also defines mixers and translators which are application-level gateways operating on top of RTP that insert, delete, or modify an RTP packet's encoding. For example, a mixer can be placed near a low-speed link to translate a high-bandwidth audio encoding to a lower-bandwidth one and forward the lower-bandwidth stream across the low-speed link instead of forcing everyone to use the lower-bandwidth, reduced-quality audio encoding. Translators include gateways that receive multicast data and forward the data using unicast to non-multicast receivers or provide translation between a group of hosts speaking only UDP/IP and a group of hosts that understand only ST-II.

RFC2250 [26] defines a payload format and packetization scheme to transport MPEG-1 and MPEG-2 video and audio streams using RTP. It defines a packetization

scheme for both MPEG-2 Transport Streams (TS) and Elementary Streams (ES).

Two main types of elementary streams are defined: audio and video. MPEG-2 Transport Streams were intended for "noisy" environments such as broadband networks. A Transport Stream consists of transport packets with a fixed-length of 188 bytes. This length was chosen with ATM and AAL-1 as a possible transmission method in mind. A transport packet of 188 bytes maps exactly into the payload of four ATM cells. However, in IP networks, multiple TS packets can be encapsulated in larger IP/UDP packets. The TS packet headers add unnecessary overhead when Transport Streams are transported over IP networks. Thus, it is more efficient to transport Elementary Streams than Transport Streams in IP networks.

This section will examine the encapsulation of MPEG video Elementary Streams. It has been shown in [6] that the packetization of elementary streams is more robust since it is packetized only once. In the case of Tranport Streams, it is packetized three times: it is firstly packetized in Packetized Elementary Stream (PES) packets then into Transport Stream (TS) packets and finally in RTP packets. Packetizing TS packets does not allow it to take into account the ES structure to reduce dependencies among packets and to maximize the amount of decodeable data at the receiver.

MPEG video ES data is not encapsulated directly in UDP packets because UDP does not provide sequence numbers or any other additional information to allow for the detection of lost or out-of-sequence data. Error recovery becomes very difficult without this information and, hence, makes UDP unsuitable for encapsulating MPEG video data.

Since MPEG pictures can be large, they will normally be fragmented into packets of size less than typical WAN/WAN MTU. An MPEG Video Sequence

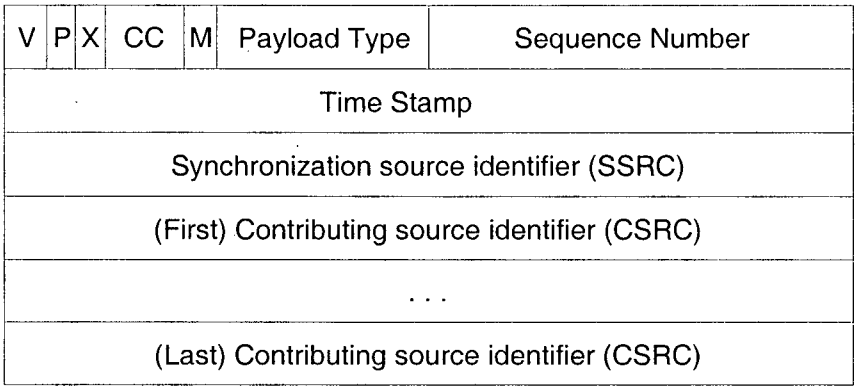


Figure 5.3: RTP Fixed Header

header, when present, is required to be at the beginning of an RTP payload. GOP and Picture headers must also be put at the beginning of an RTP payload or follow a prior header. Each ES header must be completely contained within the packet.

Each MPEG picture is made up of one or more *slices*. A slice is intended to be the unit of recovery from data loss or corruption and, therefore, an MPEG-compliant decoder will normally advance to the beginning of the next slice whenever an error is encountered in the stream. MPEG slice *begin* and *end* flags are provided in the RTP MPEG Video-specific header to facilitate this. The beginning of a slice must be either the first data in a packet after any MPEG headers or must follow after some integral number of slices in a packet. This requirement insures that the beginning of the next slice after one with a missing packet can be found without requiring that the receiver scan the packet contents.

Every RTP packet consists of a fixed header, as shown in Figure 5.3, followed by the payload data. The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The fields of the RTP fixed header have the following meaning:

Version (V): This 2 bit field identifies the version of RTP. The version defined by

RFC 1889 is two (2).

Padding (P): If this 1 bit field is set, the packet contains one or more additional padding octets at the end which are not part of the payload. The last octet of the padding contains a count of how many padding octets should be ignored, including itself. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

Extension (X): If this 1 bit field is set, the fixed header is followed by exactly one header extension, with a format defined in [52]. An extension mechanism is provided to allow individual implementations to experiment with new payload-format-independent functions that require additional information to be carried in the RTP data packet header. Additional information required for a particular payload format should **not** use this header extension, but should be carried in the payload section of the packet as profile-specific extension to the fixed header.

CSRC count (CC): This 4 bit field contains the number of CSRC identifiers that follow the fixed header.

Marker (M): For MPEG ES encapsulation, this 1 bit field is set to 1 on a packet containing an MPEG frame end code, 0 otherwise.

Payload Type (PT): For MPEG ES encapsulation, this 7 bit field is assigned a value of 32 for video elementary streams.

Sequence Number: This 16 bit field packet sequence number allows the receiver to detect packet loss and restore packet sequence.

Timestamp: For MPEG ES encapsulation, this 32 bit field contains a 90KHz timestamp representing the presentation time of the MPEG picture. It is the same for all the packets that make up a picture. It may not be monotonically increasing in video stream if B pictures are present in stream. For packets that contain only a video sequence and/or GOP header, the timestamp is that of the subsequent picture.

SSRC: This 32 bit field identifies the source of an RTP packet stream. It is a randomly generated 32 bit scalar that is meant to be unique within a multipeer session.

CSRC: This 32 bit field identifies a contributing source for the payload contained in this packet. CSRC identifiers are inserted by mixers.

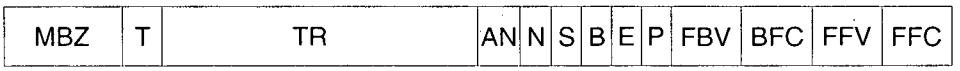


Figure 5.4: RTP MPEG Video-specific Header

Following the ALF design principle. an MPEG Video-specific header is defined which contains additional information that is specific to this payload format. The MPEG Video-specific header, shown in Figure 5.4, is attached to each RTP packet after the RTP fixed header. The fields of the MPEG Video-specific header have the following meaning:

MBZ: This 5 bit field is unused and must be set to zero in the current specification.

T: This 1 bit field indicates when an MPEG-2 specific header extension is present. This extension is a 32-bit field that contains MPEG-2 Picture header information allowing for improved error resilience; however, its inclusion in an RTP packet is optional.

TR: This 10 bit field indicates the temporal reference (the display order) of the current picture within the current GOP. This value ranges from 0 - 1023 and remains constant for all RTP packets that belong to a particular picture.

AN: This 1 bit field indicates whether the following field (N) is used to signal changes in the picture header information for MPEG-2 fields. It must be set to 0 for MPEG-1 paylods or when the N bit is not used.

N: This 1 bit field indicates a new picture header and is used for MPEG-2 payloads when the previous bit (AN) is set to 1. Otherwise, it must be set to zero. This bit is set to 1 when the information contained in the previously transmitted picture headers cannot be used to reconstruct a header for the current picture. This happens when the current picture is encoded using a different set of parameters than the previous pictures of the same type.

S: This 1 bit field indicates the presence of an MPEG Video Sequence header.

B: This 1 bit field (Begin-of-slice) is set when the start of the packet payload is a slice start code, or when a slice start code is preceded only by one or more of a Video Sequence, GOP, and/or a Picture header.

E: This 1 bit field (End-of-slice) is set when the last byte of the payload is the end of an MPEG slice.

P: This 3 bit field indicates the picture type for the current MPEG picture. I (1), P (2), B (3).

FBV: full_pel_backward_vector. This field is not used in MPEG-2 and is set to '0'.

BFC: backward_f_code. This field is not used in MPEG-2 and is set to '111'.

FFV: full_pel_forward_vector. This field is not used in MPEG-2 and is set to '0'.

FFC: forward_f_code. This field is not used in MPEG-2 and is set to '111'.

5.3.2 RTP Payload Format for Reed-Solomon Codes

A RTP payload format for forward error correction of media encapsulated in RTP using Reed-Solomon codes has been proposed in [51]. The format is generic such that it can protect any media type encapsulated in RTP. The RTP media packets being protected are not modified by the FEC operation and they are sent in a separate stream from the FEC packets. Since the FEC packets are sent as a separate stream, it is backwards compatible with non-FEC capable hosts, so that receivers which cannot understand FEC can discard the FEC packets and still receive the media packets.

The media packets are grouped into blocks of K. K can vary from one media block to the next. The Reed-Solomon coding is performed on a media block to obtain $N - K$ FEC packets which protect the K media packets. A receiver needs to receive any K of the N media or FEC packets in order to recover the K media packets.

The fields of the RTP fixed header for the FEC packets have the following meaning:

Version (V): The version field is set to 2.

Padding (P): The padding bit is computed via the protection operation described below.

Extension (X): The extension bit is computed via the protection operation described below.

CSRC count (CC): The CSRC count is computed via the protection operation described below. The CSRC list is never present, independent of the CSRC count value.

Marker (M): The marker bit is computed via the protection operation described below.

Payload Type (PT): The payload type is obtained through out of band signaling. The signaling protocol is responsible for establishing a symbol length (s) to be associated with the payload type value. For our implementation, payload type 98 is used for FEC packets. It is also associated with an 8-bit symbol length for the Reed-Solomon coding.

Sequence Number: The FEC packets have their own sequence number space. The sequence number is one higher than the sequence number in the previously transmitted FEC packet.

Timestamp: When the FEC packet is sent, the value of the media RTP timestamp is used as the timestamp of the FEC packet.

SSRC: The SSRC value should generally be the same as the SSRC value of the media stream it protects.

Following the RTP fixed header is the Reed-Solomon header as shown in Figure 5.5.

SN Base: This 16-bit field contains the sequence number of the first media packet in the media block protected by FEC.

Length Recovery: This 16-bit field is used to determine the length of any recovered packets. It is computed via the protection operation applied to the 16 bit

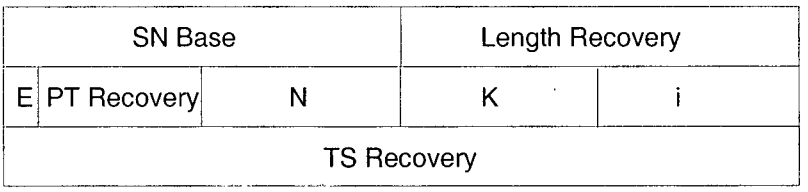


Figure 5.5: Reed-Solomon FEC Header

natural binary representation of the lengths (in bytes) of the media payload, CSRC list, extension, and padding of media packets in the media block. This field allows for the FEC procedure to be applied even when the lengths of the media packets are not identical.

Extension (E): This 1-bit field indicates a header extension. This field is currently not used and must be set to zero.

PT Recovery: This 7-bit field value is obtained via the protection operation applied to the payload type values of the media packets in the media block.

N: This 8-bit field is the total number of FEC and media packets in the coding block, minus 1.

K: This 8-bit field is the number of media packets in the media block, minus 1.

i: This 8-bit field indicates that this is the $i+1$th FEC packet of the $N-K$ FEC packets in the coding block.

TS Recovery: This 32-bit field is computed via the protection operation applied to the timestamps of the media packets in the media block.

The protection operation involves taking each media block consisting of K media packets to create K media binary arrays by appending the following fields from the header and payload together:

- Padding (P) bit (1 bit)
- Extension (X) bit (1 bit)
- CSRC Count (CC) (4 bits)
- Marker (M) bit (1 bit)
- Payload Type (PT) (7 bits)
- Timestamp (32 bits)
- Length of the CSRC list, header extension, payload, and padding of the media packet (16 bits)
- CSRC list (variable)
- Header Extension (variable)
- Payload (variable)
- Padding (variable)

If the lengths of the media binary arrays are not equal, they are padded with zeros to be the length of the longest media binary array. If the resulting media binary arrays have a length which is not a multiple of the symbol length, they are all padded further until they are a multiple of the symbol length.

The Reed-Solomon encoding operation is then applied to the K binary arrays, generating $N - K$ FEC arrays. Each FEC array is used to generate a single FEC packet. For example, if the symbol length is 8 bits, a single octet from each of the K binary arrays is taken to form a data block which the encoder uses to compute

$N - K$ parity octets. Each of the parity octets is taken to build one of the $N - K$ FEC binary arrays.

The FEC packets are created by writing the first bit in the FEC binary array into the Padding bit field of the FEC packet. The second bit in the FEC binary array is written into the Extension bit field of the FEC packet. The next 4 bits in the FEC binary array are written into the CC field of the FEC packet. The next bit in the FEC binary array is written into the Marker bit field of the FEC packet. The next 7 bits in the FEC binary array are written into the PT Recovery field of the Reed-Solomon header. The next 32 bits in the FEC binary array are written into the TS Recovery field of the RS header. The next 16 bits are written into the Length Recovery field of the RS header. The remaining bits are written to the payload of the FEC packet.

When media packets are lost during transmission, the receiver can begin reconstruction when any K of the packets (media or FEC) from the coding block have arrived. The reconstruction procedure involves creating media binary arrays from the media packets in the same manner as during the RS encoding procedure. FEC binary arrays are also re-created from the FEC packets by extracting the various fields in the same order they were placed into the FEC packet. The Reed-Solomon decoding operation is then performed on the binary arrays. This will result in N binary arrays, one of which is the recovery array corresponding to the packet to be recovered. The various bits in the recovery array are taken to create the lost packet recovering both the header and payload of an RTP packet.

A general purpose Reed-Solomon encoding and decoding package by Phil Karn [32] was modified and incorporated into the CMFS server and off-line client utility. An 8-bit symbol length was used to allow efficient processing of the data

octets. With an 8-bit symbol length, the codeword length (N) would be 255. For the MPEG-2 video streams used in our testing, the largest frames (I-frames) are encapsulated in around 40 RTP media packets. With $K = 40$, the RS encoding would produce 215 parity bytes, which translates into 215 FEC packets. This amount of protection is more than is required. Thus, shortened RS codes were employed to reduce the codeword length to produce the desired number of parity packets.

5.3.3 Media Transport (MT) Protocol

The CMFS system uses its own Media Transport (MT) protocol. It is a simple unreliable stream protocol which utilizes an underlying datagram protocol such as UDP for the transmission of data. MT is intended for the network transmission of continuous media data. This data is transmitted as a series of datagram packets which are byte sequenced so that the receiving end can detect missing data. When RTP is used in conjunction with MT, this is somewhat redundant information as RTP uses packet sequence numbering to detect packet loss. However, MT detects the amount of data lost rather than the number of packets lost.

The MT protocol does not have any error control schemes and, therefore, does not retransmit lost packets since this is usually not possible in real-time systems. The MT protocol also does not perform flow control as it is performed by CMFS at the application level.

MT runs on top of UDP to make use of its multiplexing and data checksum services. This is in addition to the basic networking services provided by the underlying IP layer. Currently, the maximum UDP/IP datagram size is set so that it can be sent in one Ethernet frame (i.e. size $<$ 1500 bytes) to avoid fragmentation. Datagrams larger than the Ethernet Maximum Transfer Unit (MTU) size of 1500

bytes will be divided into fragments. If any of the fragments is lost during network transmission then the entire packet is lost. It is therefore expected that packets larger than the MTU size will have a higher packet loss rate than packets smaller than the MTU size.

5.4 Data Collection and Analysis

The data collection for the experiments takes place primarily at the application level of the client. It was sufficient to perform the data collection at the application level because the focus of the project is on IP packet loss which can be detected and measured from the application-level RTP sequence numbers.

We implemented a simple data-logging video client for off-line error analysis. The client in our testbed network runs on an HP Vectra 200 Mhz Pentium MMX PC. The client is designed to request an MPEG-2 video stream from the CMFS server and continuously call *CmfsRead()* to read each RTP packet received. After each CmfsRead() call, a timestamp is immediately taken and saved to a log file along with the RTP packet and its length even if the packet arrives late. Similarly, the CMFS server saves each RTP packet transmitted and its length to a log file. When the FEC option is enabled, the FEC RTP packets are saved in the log at the server and client. With both the client and server log files, the packets that are not received by the client can be found in the server log file which can then be analyzed to determine what MPEG-2 video ES data units (i.e., picture header, slice, GOP header) were lost. The client log file is also parsed to measure the data units that were successfully received. With better insight into the probabilities of loss of the various data units, appropriate error control techniques can be developed to minimize the video quality degradation.

An off-line client utility was implemented to parse the client log file in order to extract the MPEG-2 video stream data contained in the payload of each RTP packet. The data is saved to a file which resembles the original MPEG-2 video ES file but without the data contained in the lost packets. When FEC is enabled, the utility attempts to recover lost packets using the FEC packets. The timestamp is used to discard the MPEG-2 ES data contained in any RTP packet that arrives later than the time at which it is supposed to have been taken from the buffer to be decoded.

The transmitted MPEG-2 video elementary streams are decoded using a software decoder [20] and saved as graphic images for video quality analysis. The video quality analysis is performed using objective and subjective quality measurement techniques. The objective video quality measurement employed in our experimental analysis is the widely used PSNR metric. PSNR provides accurate frame-to-frame pixel value comparison. Each frame received is compared to the original frame when the PSNR value is calculated for a frame. Some frames may become undecodeable, such as when the picture header is missing, when packet loss occurs. As a result, the temporal alignment (frame-to-frame sychronization) may be lost between the transmitted and original stream. A blank frame is substituted for each missing frame. No error concealment techniques are employed such as repeating the previous frame. Therefore, the video quality analysis performed gives a worst-case measurement and a baseline measurement to which error concealment techniques employed can be compared to. The final objective video quality measurement is an average PSNR value calculated by averaging all the individual PSNR values for each frame in the video sequence.

An informal subjective video quality assessment is performed using the five-

point grading scale (see Section 3.2.1). A REALmagic Hollywood2 MPEG-2 hardware decoder is used to view the transmitted video streams for subjective video quality assessment. The final subjective video quality measurement is a mean opinion score (MOS) which is an average of the viewer ratings.

Chapter 6

Experimental Results

To study the effects of packet loss on MPEG-2 video in an IP-based network, we performed several experiments using MPEG-2 video streams. In this chapter, we discuss the results from these experiments. We start with a description of the network and background traffic used in the experiments. After briefly discussing the MPEG-2 streams used in the experiments, we proceed to discuss the results of the experiments.

6.1 Network Model

The network of our experimental VoD system consists of a 3Com SuperStack 3000 10/100 Mbps Ethernet switch and a PC-based router which is an HP Vectra 200 Mhz Pentium Pro PC running FreeBSD 2.2.5. The switch supports virtual LANs (VLANs), allowing a switch to serve multiple subnets for routing tests. Three VLANs have been created and connected using the router as shown in Figure 6.1. The video server and a host, which is used as a background traffic generator source, are placed on one subnet. The video client and another host computer, which is

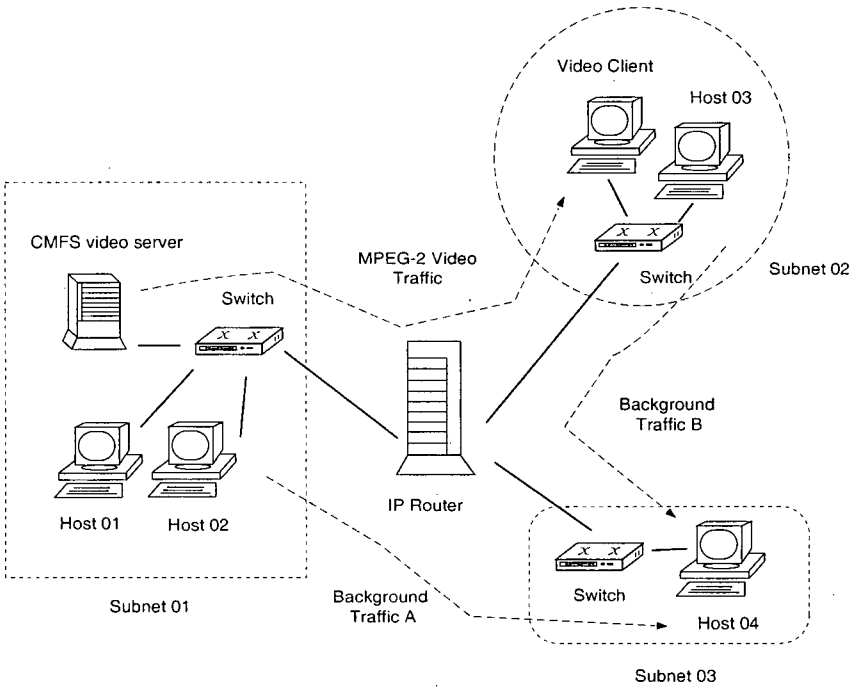


Figure 6.1: VoD Test Network

used as the background traffic sink, are placed on the second subnet. Another host on a third subnet is used to generate background traffic through the router. All the machines are connected via Fast Ethernet connections.

In our experiments, however, we utilized only two subnets. The use of the third subnet to generate cross traffic resulted in unforeseen performance degradation of the PC-based router. The throughput of the router decreases significantly whenever two or more network interfaces in the router receive data concurrently.

It was determined that the decreased throughput is a result of ethernet frames being dropped at the receive network interfaces of the router. This was confirmed after modifications were made to the network interface card (NIC) driver to collect the statistics gathered by the internal NIC counters. The network interface statistics that were gathered showed that *Receive Overrun Errors* were occurring whenever

two NICs were receiving data concurrently. These errors occur when frames are known to be lost due to the local system bus being unavailable [28].

The packets dropped at the receive interfaces are not due to insufficient router horsepower, but rather of the contention of the PCI bus. The Intel EtherExpress NICs that we are using in the router only have a 6 KB onboard buffer (3 KB receive / 3 KB transmit) and it would appear that the cards only have enough buffer space for at most two packets before the NIC starts dropping packets. The throughput of the PCI bus is 133 Mbps which should be sufficient, but when it has to start passing the bus back and forth between two or more NICs, inefficiencies will most likely mount up. Some NICs are also limited to 64 bytes per busmastering operation, leading to 24 separate busmaster operations per 1514 byte ethernet frame. So with 3 NICs and a SCSI controller running on the router, a considerable amount of time will be lost due to bus arbitration which will decrease the efficiency of the PCI bus. A possible solution is to use server-oriented NICs which have bigger buffers and/or dedicated I/O processors. However, this solution was never tested due to cost and unknown availability of drivers for the FreeBSD operating system.

The background traffic sources generate UDP packets based on the ON-OFF traffic model, as shown in Figure 6.2. Netperf [27] is used as the background traffic generator. Netperf is a benchmark program that can be used to measure various aspects of networking performance. It is capable of transmitting a stream of UDP packets using an ON-OFF model. The ON-period is specified by the number of packets to be sent in the period using the *burst size* option. The OFF-period is specified by the number of milliseconds to remain idle between bursts using the *wait time* option. The ON- and OFF-periods are constant throughout a trial run.

An attempt was made to implement our own traffic generator that generated

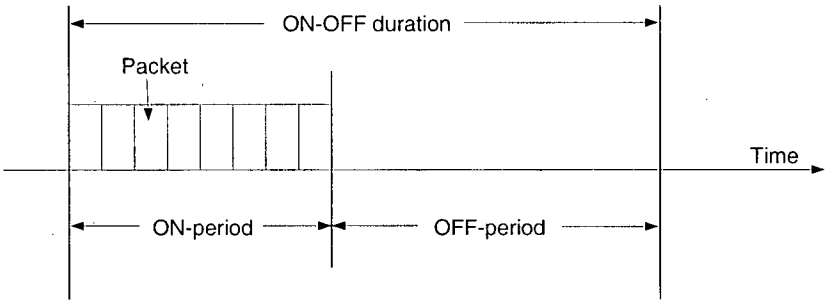


Figure 6.2: ON-OFF Traffic Model

traffic using mathematical distributions such as exponential or pareto. However, it was not possible for accurate scheduling of interpacket departure times due to the scheduling policy of the operating system. FreeBSD is not a real-time operating system and, therefore, does not guarantee when a process is scheduled to run. This becomes a significant problem when specifying a high rate of traffic generation where the interpacket departure times are small ($< 10ms$). After sending a packet, the traffic generator process would be put to sleep for a small duration. However, the process does not run immediately after it wakes up and must wait if there are other processes ahead of it in the operating system's *Ready* queue. Thus, we did not pursue the implementation of our own traffic generator any further.

The question that arises is whether the use of a simplified network topology, as shown in Figure 6.3, with the video stream and the background traffic flowing over a single link, and a constant ON-OFF traffic model will produce results that are meaningful in comparison to the Internet or an intranet. The focus of this project is on the effects of packet loss on MPEG-2 video. Therefore, it is the packet loss rates and packet loss distribution which are the important network characteristics that need to be considered.

The simplified network topology limits the competing traffic that would oc-

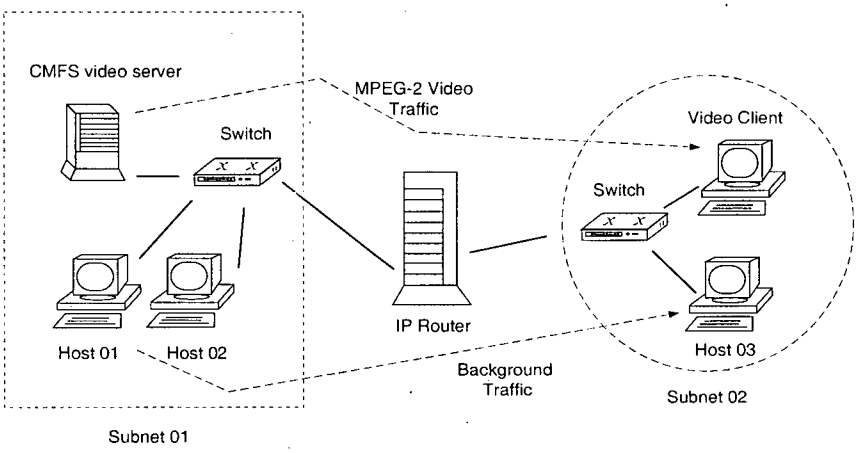


Figure 6.3: VoD Network used for Experimental Testing

cupy the network interface's output queue and, therefore, limit the maximum packet loss rate that can be observed in our testbed network. However, it will be shown that the packet loss rates needed for acceptable video quality is below the maximum packet loss rate that can be generated on our testbed network, and significantly below the packet loss rates observed on the Internet.

The packet loss distribution observed on our testbed network is similar to that observed on the Internet. A study on audio packet loss on the Internet [8] reported that the number of consecutively lost packets is small especially when the network load is low or moderate. These results are also in agreement with previous experimental results [7] obtained with non-audio UDP packets over many different connections in the Internet. We analyzed the packet loss events for each video stream transmitted under low, medium, and high network loads. Figure 6.4 shows the packet loss burst size distribution. Each line represents a set of trial runs under the same network load conditions. The packet loss events observed on our testbed network coincide closely with that observed on the Internet. All three sets of trial runs exhibit similar packet loss burst size distributions. About 70 - 80% of the loss

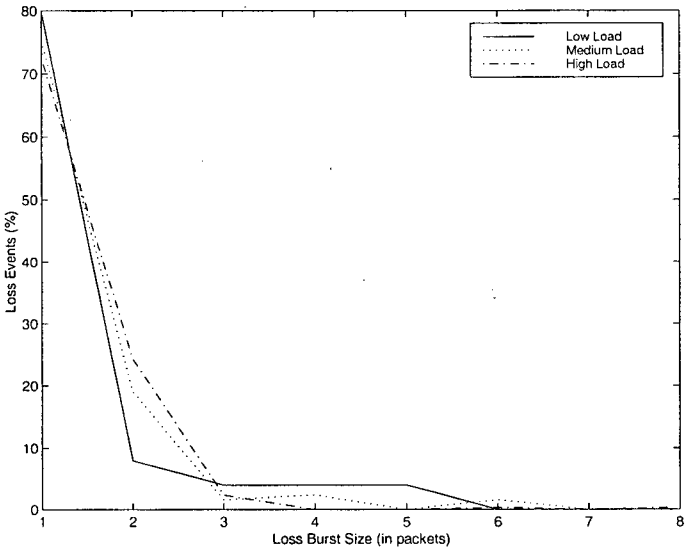


Figure 6.4: Distribution of Loss Bursts

events are single packet losses. Losses of two consecutive packets account for 8 - 24% of the loss events.

Other studies [47, 11, 10] have shown that packet losses are not independently distributed. Packet loss events are found to be correlated. The correlation of packet loss events can be seen in the remaining 20 - 30% of loss events that were not individual packet losses. In [47], Paxson notes that the pattern of loss bursts observed might be the result of the *drop-tail* queueing used in the majority of routers. With drop-tail queueing, a router queues incoming packets until the available buffer space is exhausted, and then drops any additional packets that arrive until sufficient buffer space becomes available again.

Researchers have developed loss-smoothing mechanisms, such as the Random Early Drop (RED) [19] queueing algorithm. Under drop-tail queueing, large bursts of packets are lost when a flight of closely-spaced packets arrive at a router with no available buffers. RED attempts to spread out the losses over time by dropping

incoming packets before all of the buffers have been exhausted. The packet drops are made with probabilities reflecting the proportion of the router's resources used by the connection, so the policy is fairer than the drop-tail queueing policy.

6.2 MPEG-2 Video Streams

Three video streams which represent different levels of activity (motion) and content were used in our experiments. The video sequences were obtained from web sites and are known as *Cheer*, *Ballet*, and *Susi*. Cheer consists of a group of cheerleaders performing their team cheer at a game. Ballet consists of two ballet dancers dancing on a bare stage. Susi consists of a woman talking on the telephone. Cheer can be classified as high activity and high spatial detail, Ballet as medium activity and medium spatial detail, and Susi as low activity and low spatial detail. For brevity, video-H, video-M, and video-L will be used to refer to Cheer, Ballet, and Susi, respectively.

All three video streams are MPEG-2 Video Elementary Streams consisting of 300 frames with a picture size of 704 by 480 pixels and encoded at a constant bit rate (CBR) of 6 Mbps with a frame rate of 30 frames/s, which conforms with the broadcast quality Main Profile@Main Level CCIR 601 video. The streams have been encoded using the IBBPBBP format where I, B, and P correspond to I-, B-, and P-pictures respectively. In particular, they have been encoded with N=15 and m=3, where N is the number of pictures in the group-of-pictures (GOP), and m is the distance between two P-pictures. Table 6.1 provides additional statistics of the MPEG-2 video streams.

	Frames	Slices	Bytes			Packets		
			video–H	video–M	video–L	video–H	video–M	video–L
I	21	630	866782	664211	864736	843	660	880
P	80	2400	2458287	2983630	2854488	2374	2701	2472
B	199	5970	2930873	2598643	2532044	3415	3114	3037
Total	300	9000	6255942	6246484	6251268	6632	6475	6389

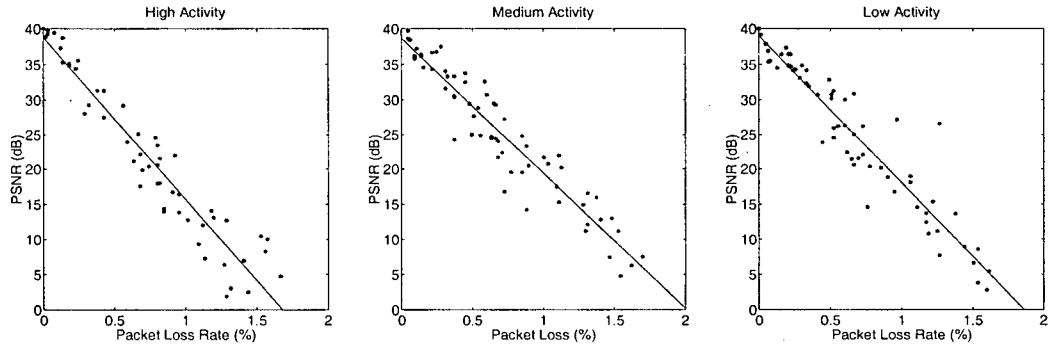
Table 6.1: Video Stream Statistics

6.3 Experimental Results

In this section, we discuss the results from the experiments we performed to study the effects of packet loss on the quality of the MPEG-2 video. All the experiments were performed using the network topology of Figure 6.3 and Netperf as the traffic source generator. The three MPEG-2 video streams were transmitted across the network under a number of different network loads in order to obtain packet traces for a range of packet loss rates.

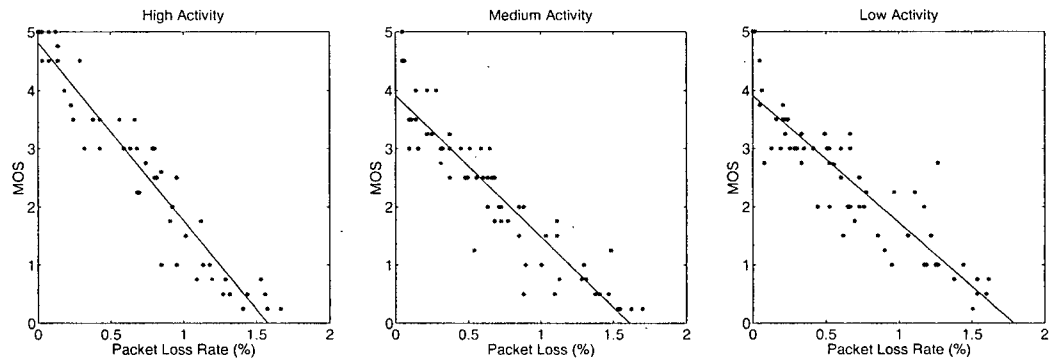
Our initial experimental results showed that even low packet loss rates will severely degrade the video quality. Packet loss rates above 1.5% resulted in video that was considered very annoying and unwatchable. Previous work on the transmission of MPEG-2 video has primarily been over ATM networks. With ATM networks, ATM cells are small compared to IP packets. If an ATM cell is lost, only one or a few macroblocks of a slice are lost while an IP packet loss may result in one or more slices being lost. For ATM, generally unacceptable video quality is produced with cell loss rates of 0.1% [5].

The objective PSNR video quality assessment measurements for video-H, video-M, and video-L are plotted in Figure 6.5. A least-squares fit line has been added to each scatter plot. We see that, for the three different activity streams, the PSNR video quality assessment is linearly correlated with the packet loss rate.



(a) (b) (c)

Figure 6.5: Packet Loss vs. Objective PSNR for (a) Video-H, (b) Video-M, and (c) Video-L



(a) (b) (c)

Figure 6.6: Packet Loss vs. Subjective MOS for (a) Video-H, (b) Video-M, and (c) Video-L

In fact, they have very similar slopes which would indicate that the video activity and content have no effect on the objective PSNR video quality metric under different packet loss rates. The PSNR metric is based on a per pixel comparison of the transmitted frame and the corresponding original frame. The PSNR metric examines each frame in isolation and does not measure pixel value differences between consecutive frames which would be necessary to account for the motion in a video sequence. Thus, the activity level of the video does not play a role in the objective PSNR video quality measurement.

The subjective Mean Opinion Score (MOS) video quality assessment measurements for the high, medium, and low activity streams are plotted in Figure 6.6. A least-squares fit line has been added to each scatter plot. Video-H has a greater slope than the other two activity streams which have similar slopes. A greater slope would indicate that the video quality is degrading at a faster rate. However, it is misleading as the least-squares fitting line for video-M and video-L interpolate an MOS value of 3.9 at a packet loss rate of 0% which is incorrect. A packet loss rate of 0% should indicate an MOS value of 5 since there is no difference in the picture between the original and the received stream. The subjective MOS video quality measurements exhibit large variations and do not show tight linear correlation. The variations indicate that the lower activity streams require lower packet loss rates to degrade the subjective video quality.

The large variation exhibited in the plots are primarily due to the subjective nature of the measurement method. There are many human processes that are not clearly understood which would have an influence in a viewer's assessment for the overall quality of a video sequence. It is not clear how a viewer will react to a quality variation characterized by a certain percentage of distorted scenes or whether

a viewer's assessment depends upon the distribution of the individual distortions, i.e., their magnitude and duration. It has also been shown in [37] that human memory processes play an extremely important role in the perception of picture quality, and furthermore, the perception of quality reported by a view during a program is dependent upon when the reporting takes place. Their results showed the phenomenon, termed *recency* by psychologists, has a significant influence in the viewer's video quality assessment. Recency is an effect which acts to emphasize memory events which have occurred within the span of the working memory ($\sim$15 seconds) before they are consigned to longer-term memory processes. The viewers in their study perceived the video quality to be better when distortions in the video occurred in the beginning, as compare to when they occurred at the end, because they had largely forgotten about the distortion.

Video-H shows greater resilience to packet loss rate in the range between 0 - 0.2% according to the subjective MOS measurements. It maintains an MOS rating of at least 4 which indicates perceptible video impairments but which are not annoying. We see that video-L and video-M have greater variability in this packet loss rate range with the quality rating dropping down to 3 which is slightly annoying to the viewer.

Type	**Description**
Tiling or pixelation	Formation of small blocks with distinct boundaries
Screen blanking	Loss of video (black screen), flickering effect
Motion jerkiness	Irregular or unnatural motion observed
Frame freezing	Screen freezes (similar to motion jerkiness but of longer duration)
Error blocks	Small solid-colour blocks appear

Table 6.2: Description of digital video artifacts

Digital video exhibits a list of unique impairments that are not encountered in

Quality Boundary	Packet Loss (%)			PSNR (dB)		
	video–H	video–M	video–L	video–H	video–M	video–L
5 - 3	0.0–0.6	0.0–0.4	0.0–0.3	40.0–24.9	40.0–32.0	40.0–31.0
3 - 2	0.6–0.9	0.4–0.7	0.3–0.8	24.9–17.5	32.0–24.0	31.0–22.0
2 - 0	> 0.9	> 0.7	> 0.8	< 17.5	< 24.0	< 22.0

Table 6.3: Subjective–to–Objective Quality Mapping

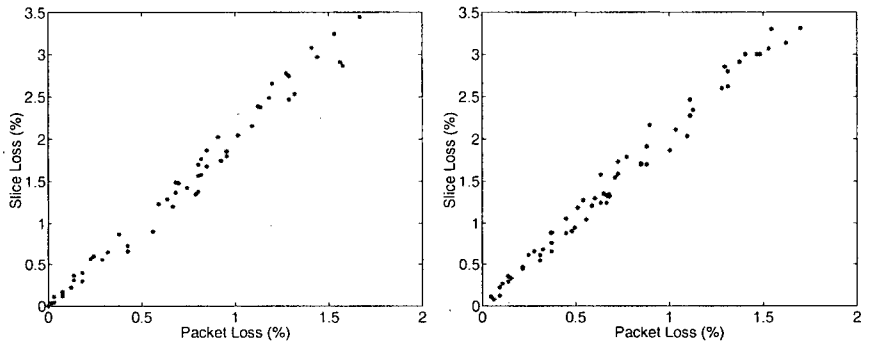
analog video. Some of the digital video artifacts include tiling or pixelation, screen blanking, motion jerkiness, frame freezing, and error blocks. Table 6.2 describes these digital video artifacts. Packet loss can result in tiling, error blocks, motion jerkiness, and screen blanking. The artifacts observed vary from decoder to decoder depending on the level of error concealment implemented. The software decoder used in this project does not perform any error concealment.

An MOS rating between 5 and 3 would be considered acceptable video quality with only slightly annoying digital video artifacts such as tiling and error blocks typically due to slice loss. An MOS rating between 3 and 2 would be video that would be considered annoying but bearable to watch. The viewer would observe occasional motion jerkiness and screen blanking in addition to the tiling and errors block artifacts. With an MOS rating below 2, the video is extremely annoying and unbearable to watch due to the extreme flickering when multiple frames are lost and undecodeable.

We can obtain a subject-to-objective quality mapping by using the packet loss rate as the common variable. The packet loss rate can be determined for a particular MOS value from Figure 6.6 which can then be used to map back in to the objective PSNR quality domain in Figure 6.5. For example, an MOS value of 3 corresponds to a packet loss rate of 0.6% which maps into a PSNR value of 24.9. A complete subjective-to-objective quality mapping is given in Table 6.3.

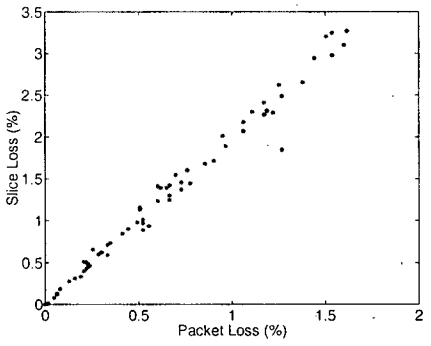
From Table 6.3, the MOS value of 3 corresponds to a PSNR of 24.9 for video-H, 32.0 for video-M, and 31.0 for video-L. These objective results appear to be in conflict with the subjective assessments. Video-H with a lower PSNR value than video-M and video-L clearly shows that it has greater distortion since the PSNR metric is a comparative metric that only computes pixel differences. The subjective assessments show that the viewer perceives the picture quality to be the same. Figure 6.5 shows the three different activity streams have a similar slope and, therefore, the video quality degradation should occur at the same rate. However, Figure 6.6 shows that video-H is more resilient to packet loss. As shown here and in other literature, the objective PSNR video quality metric is poorly correlated with human perception. The PSNR metric does not take the visual masking phenomenon into consideration [57]. Every single errored pixel contributes to the decrease of the PSNR value, even if this error is not perceived by the viewer. It appears that an increase in the temporal frequency results in a decrease in the sensitivity of the visual system. This would help explain why video-H would have the same subjective quality measurement as video-M and video-L but at the same time have a lower PSNR value.

Further evidence supporting the hypothesis that the difference in the video quality between the three activity streams is due to characteristics of the human visual system can be found by examining the lost data. Upon examining the MPEG-2 data units contained in the lost packets, all three activity streams were found to have approximately equal percentages of picture header and slice losses at the same packet loss rate, as shown in Figure 6.7(a,b,c) and Figure 6.8(a,b,c). It should not be unexpected for all three activity streams to have similar percentages of picture header and slice losses at the same packet loss rate since they were encoded using



(a)

(b)

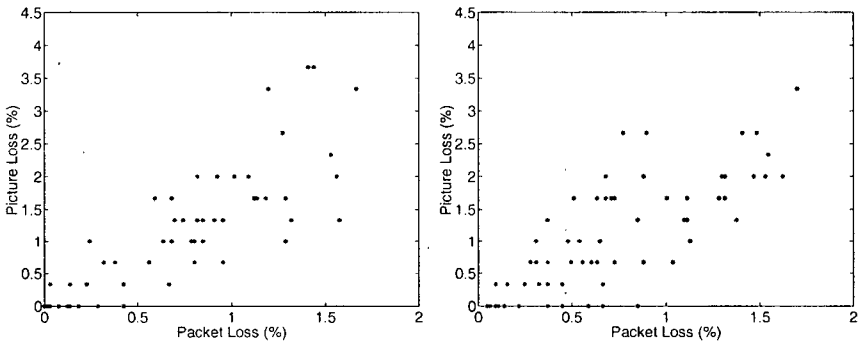


(c)

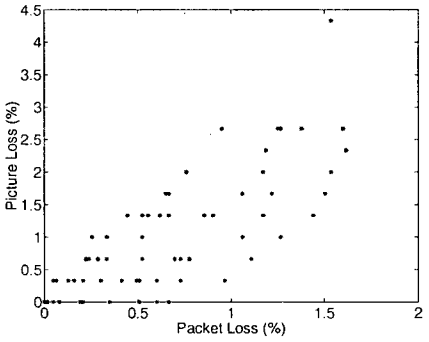
Figure 6.7: Packet Loss vs. Slice Loss for (a) Video-H, (b) Video-M, and (c) Video-L

the same encoding parameters and, therefore, the streams would have approximately the same number of MPEG video data units (i.e., headers, slices, etc.) as well as requiring approximately the same total number of packets for transmission.

We can see in Figure 6.8 that the picture header loss rate does not have very good correlation with the packet loss rate. The variability of the picture header loss rate is very high. For example, Figure 6.8(b) shows a packet loss rate of 0.85% can result in a picture header loss rate between 0% and 2.6%. On the other hand, Figure 6.7 shows that slice loss has a very strong linear correlation with packet loss.



(a) (b)



(c)

Figure 6.8: Packet Loss vs. Picture Loss for (a) Video-H, (b) Video-M, and (c) Video-L

Slice loss has a very strong correlation with packet loss because nearly every packet contains one or more slices or a portion of a slice. A rough calculation based on the stream statistics information (see Table 6.1) indicates that about 1.4 slices is contained in each packet which corresponds to the plotted data. Picture header loss is poorly correlated with packet loss because a picture header only occurs in only 300 out of a total of around 6500 packets, which is only about 4.6% of the packets. Thus, the probability of picture header loss is lower, but it increases the variability which is primarily dependent on the packet loss characteristics (i.e., distribution and

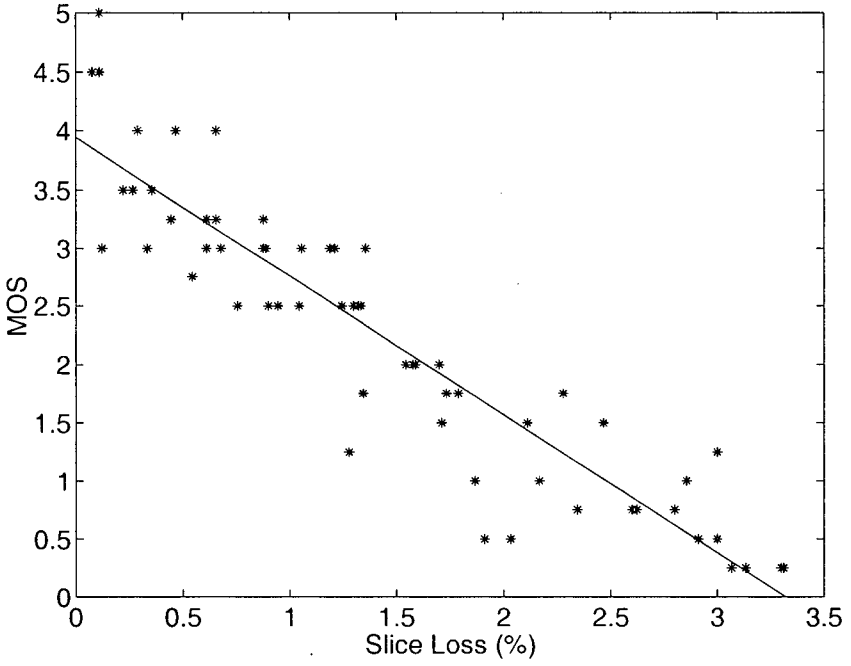


Figure 6.9: Slice Loss vs. Subjective MOS

pattern).

In addition to studying the relationship of packet loss with video quality, we examine how the loss of the MPEG data units (i.e., slices and picture headers) affect the video quality. A slice loss typically results in tiling, pixelation, and error blocks. A picture header loss results in the loss of the whole picture which is perceived as motion jerkiness and screen blanking. Better error control techniques can be developed by knowing which are the important MPEG data units that need greater protection. Error control techniques, such as those involving redundacy, that do not take the importance of the different data units into account may not be effectively and efficiently minimizing the video quality degradation, as they may increase bandwidth usage above what is needed to achieve the same video quality.

We plot slice loss and packet loss against the subjective MOS assessment

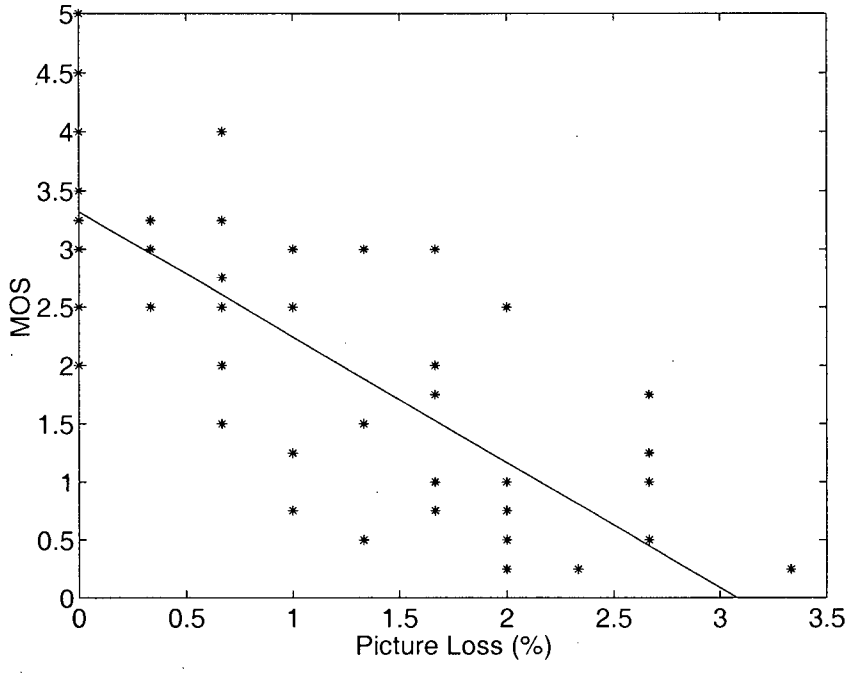


Figure 6.10: Picture Header Loss vs. Subjective MOS

measurements for video-M, as shown in Figure 6.9 and 6.10. Plots for video-H and video-L are not shown as they are similar to the video-M plots. We can see from the plots that the slice loss is more tightly correlated with the subjective MOS assessments than picture loss. It is clear that the variation of the video quality due to picture loss is much greater than that due to slice loss. Figure 6.10 shows that slices play a significant role in the video quality. With no picture losses, the slice losses can degrade the video quality to the point where it is annoying and unbearable to watch. If we examine the two trial runs, with an MOS value of 4 and 1.5, that occur at a picture loss rate of 0.67%, we find that it is the loss of slices which account for the greater video quality degradation. Both trial runs have lost 2 B-picture headers but one trial has only lost 59 slices while the other has lost 190 slices. These results show that slice loss is the dominant factor contributing

to the video quality degradation rather than picture header loss. When networking congestion increases, the video quality degrades as the number of occurrences of tiling, pixelation, and error blocks increases due to slice loss. By the time viewers perceive motion jerkiness and screen blanking due to picture header loss, slice loss has already degraded the video quality to the point where it is too annoying and unbearable to watch. Therefore, it is more important to protect slices than picture headers.

To get a better grasp of the effects of packet loss on the video quality, we use the frame error rate, as calculated in [11], which is a measure of the number of frames that contain errors. The frame error rate metric can provide a measurement that we can better relate to than PSNR and MOS. Errors are due to a loss of packets in the current frame or in a frame that the current frame is predicted from. Packet loss errors spread within a single picture up to the next synchronization point (i.e., slice). This is referred to as spatial propagation and may damage any type of picture. When loss occurs in a reference picture (I-picture or P-picture), the error will remain until the next I-picture is received. This causes the error to propagate across several non-intra-coded pictures until the end of the group-of-pictures (GOP), which typically consists of about 12 to 15 pictures. This is known as temporal propagation and is due to inter-frame prediction. Errors in B-pictures do not propagate because B-pictures are not used in the prediction of other pictures.

Figure 6.11 shows the relationship between the packet loss rate and the frame error rate for video-M. Video-H and video-L showed very similar relationships. It clearly shows that low packet loss rates translate into much higher frame error rates. For example, a 1% packet loss rate translates into a 40% frame error rate. This measurement indicates the difficulty of sending MPEG-2 video over a lossy,

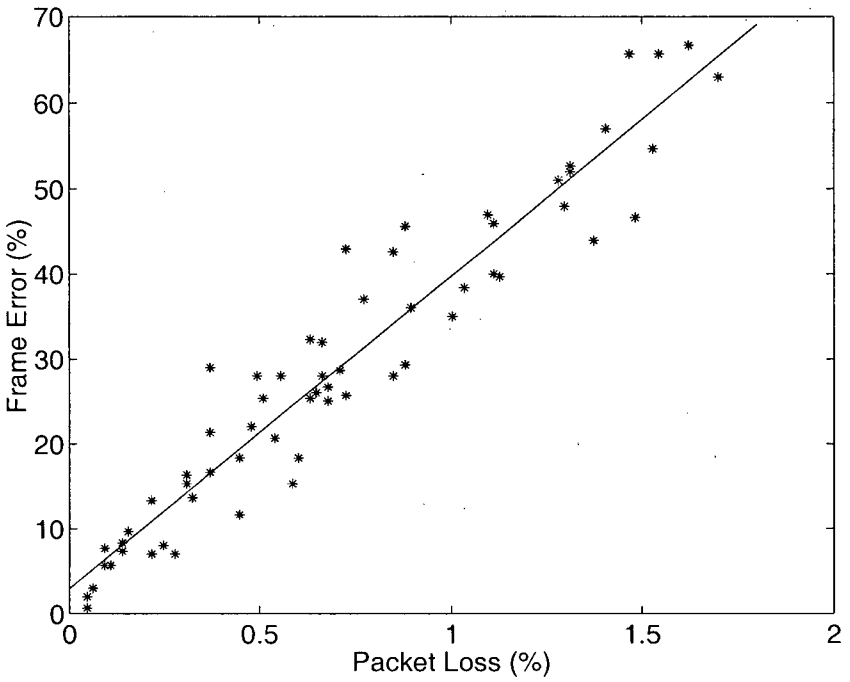


Figure 6.11: Packet Loss vs. Frame Error

best-effort IP-based network.

The experiments performed thus far have shown that packet loss rates as low as 0.8% results in unsatisfactory video quality. Various error control and concealment techniques have been proposed to improve video quality as discussed in Chapter 4. We investigate the effectiveness of forward error correction, using Reed-Solomon coding, on the video quality. We apply FEC only to the I-pictures since these are reference pictures that would propagate errors to all the pictures that follow in the Group-of-Pictures. Applying FEC to all the pictures would significantly increase the overhead and further add congestion to the network.

Using shortened RS codes, the codeword length N was set for each I-picture to provide protection for up to a 10% packet loss in an I-picture. For example, if the number of RTP media packets K required to encapsulate a picture was 30, then the

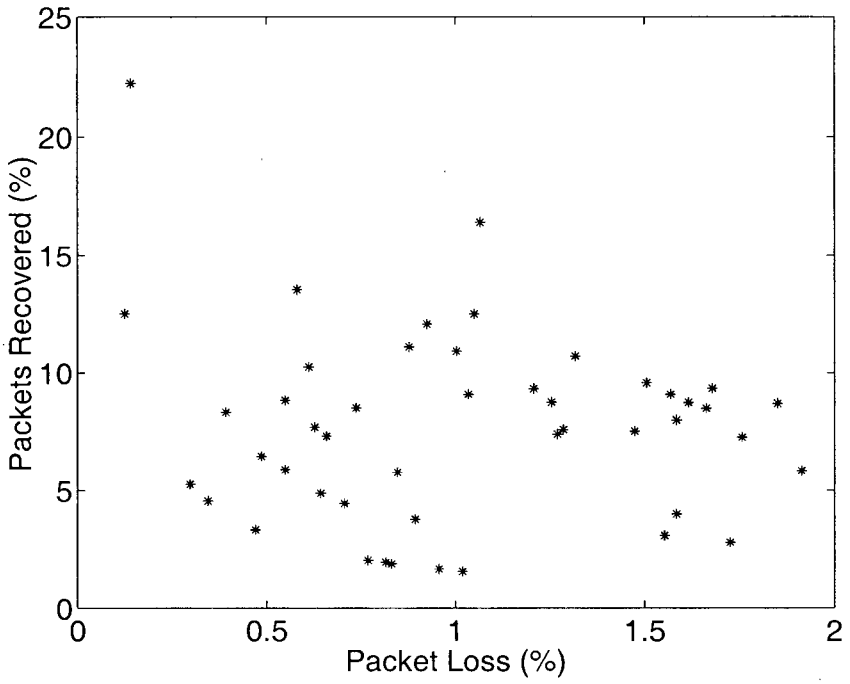
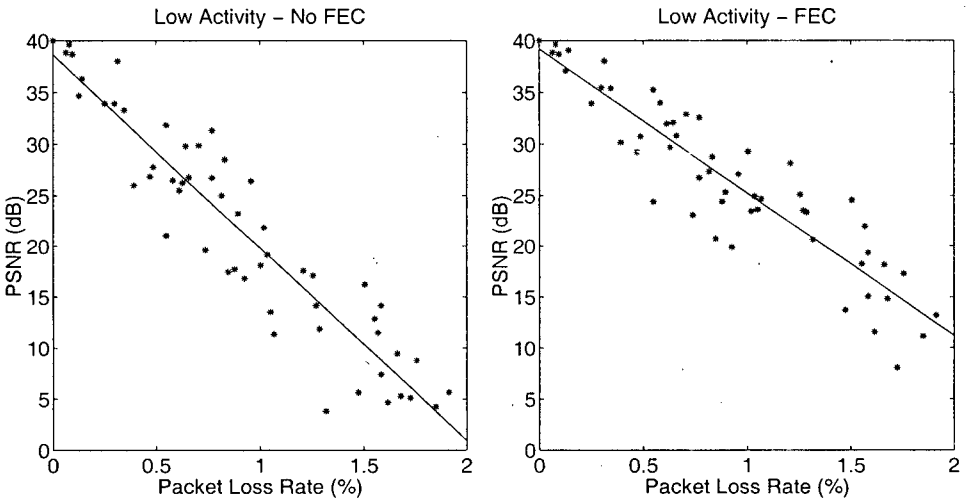


Figure 6.12: Packet Loss vs. Packet Recovery

codeword length N would be set to 33 to provide $N-K$ parity packets of protection.

For video-L, each I-picture is encapsulated in 30 to 50 RTP media packets. FEC added 3 to 5 packets of overhead to each I-picture. Recovery is possible whether the packets are lost in a burst or randomly throughout the I-picture packet block. With FEC, video-L generated 71 RTP FEC packets for the 21 I-pictures in the whole video stream. The FEC added a 1.1% packet overhead and a 1.6% byte overhead to the stream.

The FEC was able to recover all the lost packets of the I-pictures except on one test run. However, the I-picture packet losses only account for about 10% of the packet losses. Figure 6.12 shows the percentage of packets recovered out of all the packets that were lost. Greater packet losses occurred in the P-pictures and B-pictures. This would be expected given that 13.8% of the video-L packets are for



(a) (b)

Figure 6.13: Packet Loss vs. PSNR (a) without FEC and (b) with FEC

I-pictures while the P-pictures and B-pictures account for 38.7% and 47.5% of the packets, respectively.

FEC provides greater video quality improvement as packet loss rates increase as shown in Figure 6.13. Figure 6.13(a) and (b) shows the objective PSNR video quality assessment measurement without and with FEC. The video quality improved by 10 dB at a packet loss rate of 2% as compared to an improvement of 5 dB at a packet loss rate of 1%. Given that the threshold point for acceptable video quality for video-L is at 22 dB, this threshold point occurs at a packet loss rate of 0.9% without FEC, and at 1.3% with FEC. FEC increased the tolerable packet loss rate by 0.4%. An additional 1.1% packet overhead for FEC provided a 45% increase in tolerability for packet loss.

Chapter 7

Conclusions and Future Work

In this thesis, we examine the effects of packet loss on the quality of the MPEG-2 video transported over an IP network. Some of the factors that have an impact on the video quality are the packet delays, the number of packets lost, the number of pixels in an impaired region, and the type of video information lost. Video is delay sensitive and, therefore, any data that is delayed and arrives beyond the decode time is considered lost and is treated as such by the application. The amount of data lost due to an IP packet loss is a magnitude larger than an ATM cell loss. With an IP packet loss, one or more slices may be lost while an ATM cell loss results only in the loss of one or more macroblocks. Thus, the number of pixels affected in an impaired region is significantly larger when MPEG-2 video is transported over an IP network.

To study the effects of packet loss on the MPEG-2 video quality, we employed a video-on-demand testbed network. The testbed network consisted of UBC's Continuous Media File Server (CMFS), a client, and several hosts acting as the source and sink for the background network traffic, all of which were connect via Fast

Ethernet. The experiments were run under different network loads to generate a range of packet loss rate conditions. We used three 6 Mbps CBR MPEG-2 video elementary streams of differing activity levels in our experiments. In order to assess the video quality, we used the objective peak signal to noise ratio (PSNR) metric and the subjective mean opinion score (MOS) assessment.

With MPEG-2 video being a digital format, it inherits a number of unique impairments that can be observed as tiling or pixelation, error blocks, motion jerkiness, and screen blanking. Our experiments show that packet loss rates above 1.5% result in video that is considered very annoying and unbearable to watch. We confirmed that the objective PSNR video quality metric is poorly correlated with subjective quality assessments. We found that the activity level of the video influences the subjective MOS video quality assessments but not the objective PSNR quality measurements. In particular, the high activity stream showed greater resilience to packet loss in the range between 0 - 0.2%. The subjective assessments rated the high activity stream to have better quality than the medium and low activity streams. The three activity streams exhibited very similar objective PSNR measurements which would indicate that they had a similar number of errored pixels under the same packet loss rates. Upon further examination of the lost packets, we found that they all had similar percentages of lost MPEG picture headers and slices. The perceived difference in video quality was found to be attributed to characteristics of the human visual system. It appears that an increase in the temporal frequency results in a decrease in the sensitivity of the visual system. Thus, the viewer does not perceive as many of the distortions and artifacts that occur in the video sequences with high motion.

We also found that slice loss is linearly correlated with packet loss while

picture header loss is poorly correlated with packet loss. The poor correlation is a result of the small number of picture headers that occur in the stream as compared to the large number of slices. A slice loss typically results in tiling, pixelation, and error blocks. A picture header loss results in the loss of the whole picture which is observed as motion jerkiness and screen blanking. We found that slice loss is the dominant factor contributing to the degradation in video quality rather than picture header loss at packet loss rates below 1.5%. By the time viewers perceive motion jerkiness and screen blanking due to picture loss, video distortions due to slice loss have already degraded the video quality to the point where it is annoying and unbearable to watch.

We found that packet loss rates as low as 1% produced frame error rates as high as 45% due to propagation of errors. This gives a good indication of the difficulty of sending MPEG-2 video over a lossy, best-effort IP-based network. To combat the problem of video quality degradation due to packet loss, we investigated the effectiveness of forward error correction using Reed-Solomon coding. We found that FEC provided greater video quality improvement as the packet loss rate increased. The video quality improved by 10 dB at a packet loss rate of 2% as compared to only 5 dB at a packet loss rate of 1%. We found that applying FEC to the I-pictures increased the stream's packet loss rate tolerance by 45% at the boundary point for acceptable video quality with an added packet overhead of 1.1%.

Before closing, we would like to propose areas for future research. One area that requires further study is the objective video quality measurements. The results of our experiments confirm the poor correlation between objective and subjective video quality measurements. Further research is needed to develop objective video quality measurement techniques that closely correlate with subjective mea-

surements.

To improve video quality, a more agressive FEC strategy that also protects P-pictures could be used. However, the additional overhead may increase the network congestion to the point where the FEC becomes ineffective. Applying FEC to B-pictures would be less effective because errors never propagate from these pictures that appear on the screen for only $1/30^{th}$ of a second.

The size of the packets also affects the video quality. A loss of a large packet would result in a larger area of a picture being distorted. However, using small packets increases the overhead and may increase congestion in the routers that may be limited by the number of packets rather than the size of the packet. Further research is needed to determine the effectiveness of optimizing the packet size to improve the video quality.

Error concealment techniques can improve a video stream's resilience to packet loss up to a certain point. However, to guarantee acceptable video quality, some form of quality of service is necessary in order to be able to transport high-quality MPEG-2 video over an IP network.

Bibliography

[1] B. Ahn, K.-H. Cho, H. Song, J. Park, H. yoon, and J. W. Cho, "Design of Rate-based Congestion Control Scheme for MPEG Video Transmission in ATM Networks," *Proceedings of IEEE Global Telecommunications Conference '97*, Pheonix, AZ, pp. 1690-1694, November 1997.

[2] S. Aign and K. Fazel, "Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec," *Proceedings of IEEE International Conference on Communications '95*, pp. 1778-1783, June 1995.

[3] M. Andronico, A. Lombardo, S. Palazzo, and G. Schembra, "Performance Analysis of Priority Encoding Transmission of MPEG Video Streams," *Proceedings of IEEE Global Telecommunications Conference '96*, pp. 267-271, November 1996.

[4] ANSI Standard T1.801.03-1996, "Digital Transport of One-Way Video Signals, Parameters for Objective Performance Assessment."

[5] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet Loss Resilience of MPEG-2 Scalable Video Coding Algorithms," *IEEE Trans-*

actions on Circuits and Systems for Video Technology, 6(5), pp. 426-435, October 1996.

[6] A. Basso, G. L. Cash, and M. R. Civanlar, "Transmission of MPEG-2 Streams over Non-Guaranteed Quality of Service Networks," *Proceedings of Picture Coding Symposium*, Berlin, September 1997.

[7] J-C. Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," *Proceedings of ACM SIGCOMM '93*, San Francisco, CA, pp. 289-298, August 1993.

[8] J-C. Bolot, H. Crépin, and A. V. Garcia, "Analysis of Audio Packet Loss in the Internet," *Proceedings of NOSSDAV '95*, Durham, NH, pp. 163-179, April 1995.

[9] J-C. Bolot and T. Turletti, "Experience with Control Mechanisms for Packet Video in the Internet," *Computer Communications Review*, 28(1), January 1998.

[10] M. S. Borella and D. Swider, "Internet Packet Loss: Measurement and Implications for End-to-End QOS," *Proceedings of the 1998 ICPP Workshops on Architectural and OS Support for Multimedia Applications*, Minneapolis, MN, pp. 3-12, August 1998.

[11] J. M. Boyce and R. D. Gaglianello, "Packet Loss Effects on MPEG Video Sent Over the Public Internet," *Proceedings of ACM Multimedia '98*, Bristal, England, pp. 181-190, September 1998.

[12] C. Cavé, R. Ragot, and M. Fano. "Perception of Sound-Image Synchrony in Cinematographic Conditions," *Proceedings of the Fourth*

Workshop on Rhythm Perception and Production, Bourges, France, pp. 25-30, June 1992.

[13] D. D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," *Proceedings of ACM SIGCOMM '90*, Philadelphia, Pennsylvania, pp. 200-208, September 1990.

[14] P. Cuenca, L. Orozco-Barbosa, A. Garrido, F. Quiles, and T. Olivares, "A Survey of Error Concealment Schemes for MPEG-2 Video Communications Over ATM Networks," *IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, Volume 1, pp. 118-121, May 1997.

[15] P. Cuenca, A. Garrido, F. Quiles, and L. Orozco-Barbosa, "Some Proposals to Improve Error Resilience in the MPEG-2 Video Transmission over ATM Networks," *Proceedings of INFOCOM '98*, pp. 668-675, March 1998.

[16] G. M. Drury, "Picture Quality Issues in Digital Video Compression," *Proceedings of International Broadcasting Conference 1995*, Amsterdam, Netherlands, pp. 13-18, September 1995.

[17] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Loss Recovery Techniques for Transmission of MPEG Video over ATM Networks," *Proceedings of ICC/SUPERCOMM '96*, pp. 1406-1410, June 1996.

[18] D. K. Fibush, "Practical Application of Objective Picture Quality Measurements," *Proceedings of International Broadcasting Conference 1997*, Amsterdam, Netherlands, pp. 504-513, September 1997.

[19] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, 1(4), pp. 397-413, August 1993.

[20] C. Fogg, mpeg2decode/mpeg2encode, In *MPEG Software Simulation Group*, 1996.

[21] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3), pp. 238-247, June 1993.

[22] S. Gringeri, B. Khasnabish, A. Lewis, K. Shuaib, R. Egorov, and B. Basch, "Transmission of MPEG-2 Video Streams over ATM," *IEEE Multimedia*, 5(1), pp. 58-71, January-March 1998.

[23] T. Han and L. Orozco-Barbosa, "Performance Requirements for the Transport of MPEG Video Streams over ATM Networks," *Proceedings of IEEE International Conference on Communications '95*, Volume 1, pp. 221-225, 1995.

[24] M. Handley, "An Examination of Mbone Performance," *USC/ISI Research Report: ISI/RR-97-450*, April 1997.

[25] S. S. Hemami and T H.-Y. Meng, "Transform Coded Image Reconstruction Exploiting Interblock Correlation," *IEEE Transactions on Image Processing*, Volume 4, pp. 1023-1027, July 1995.

[26] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," *RFC 2250*, January 1998.

[27] IND Networking Performance Team Hewlett-Packard, "Netperf: A Network Performance Benchmark", Revision 2.1, 1996.

[28] Intel Corporation, "Intel 82557 Fast Ethernet PCI Bus Controller", Specification document, October 1996.

[29] ISO/IEC International Standard 13818, "Generic Coding of Moving Pictures and Associated Audio Information," November 1994.

[30] ITU-R BT.500, "Methodology for the Subjective Assessment of the Quality of Television Pictures."

[31] G. Karlsson, "Asychronous Transfer of Video," *Technical Report R95-14*, Swedish Institute of Computer Science, 1995.

[32] P. Karn, A General Purpose Reed-Solomon Encoder/Decoder in C, Version 2.0, http://people.qualcomm.com/karn/code/fec/rs-2.0.tar.gz, May 1999.

[33] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, and J. Mahler, "Real-Time Voice over Packet Switched Networks," *IEEE Network*, 12(1), pp. 18-27, January-February 1998.

[34] C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System," *Proceedings of the IS&T Symposium on Electronic Imaging: Science and Technology*, San Jose, CA, February 1996.

[35] C. J. van den Branden Lambrecht, "A Working Spatio-Temporal Model of the Human Visual System for Image Restoration and Quality Assessment Applications," *Proceedings of the 1996 IEEE International*

Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, pp. 2291-2294, May 1996.

[36] V. C. S. Lee, J. K. Y. Ng, K. Lam, and S. Hung, "Performance Studies of Transmitting Real-Time MPEG-I Video in ATM Networks," *Proceedings of the 1996 21st Conference on Local Computer Networks*, Minneapolis, MN, pp. 153-160, 1996.

[37] N. K. Lodge and D. Wood, "New Tools for Evaluating the Quality of Digital Television - Results of the Mosaic Project," *Proceedings of International Broadcasting Conference '96*, Amsterdam, Netherlands, pp. 323-330, September 1996.

[38] J. Lubin, M. H. Brill, and R. L. Crane, "Vision Model-Based Assessment of Distortion Magnitudes in Digital Video," Presented at the *Made to Measure '96 Symposium*, Montreaux, Switzerland, November 1996.

[39] R. Mechler, "Media Transport API," Working document, January 1997.

[40] E. Mellaney, L. Orozco-Barbosa, and G. Gagnon, "Study of MPEG-2 Video Traffic in a Multimedia LAN/ATM Internetwork System," *IEEE Transactions on Circuits and Systems for Video Technology*, 7(4), pp. 663-674, August 1997.

[41] A. Narula and J. S. Lim, "Error Concealment Techniques for an All-Digital High-Definition Television System," *Proceedings of SPIE Con-*

ference on Visual Communication Image Processing, Cambridge, MA, pp. 304-315, 1993.

[42] G. Neufeld, D. Makaroff, and N. Hutchinson, "Design of a Variable Bit Rate Continuous Media File Server for an ATM Network," *IS&T/SPIE Multimedia Computing and Networking*, San Jose, CA, January 1996.

[43] G. Neufeld, D. Makaroff, and N. Hutchinson, "Server Based Flow Control in a Distributed Continuous Media Server," *Proceedings of NOSSDAV '96*, Zushi, Japan, April 1996.

[44] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based Loss Recovery for Reliable Multicast Transmission," *Proceedings of ACM SIGCOMM '97*, Cannes, France, pp. 289-300, September 1997.

[45] S. Olsson, M. Stroppiana, and J. Baina, "Objective Methods for Assessment of Video Quality: State of the Art," *IEEE Transactions on Broadcasting*, 43(4), pp. 487-495, December 1997.

[46] M. Orzessek and P. Sommer, "ATM & MPEG-2: Integrating Digital Video into Broadband Networks," *Prentice-Hall*, 1998.

[47] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," *Ph.D. Thesis*, University of California, Berkeley, April 1997.

[48] R. S. Ramanujan, J. A. Newhouse, M. N. Kaddoura, A. Ahamad, E. R. Chartier, and K. J. Thurber, "Adaptive Streaming of MPEG Video over IP Networks," *Proceedings of 22nd Annual Conference on Local Computer Networks*, Minneapolis, MN, pp. 398-409, November 1997.

[49] M. J. Riley and I. E. G. Richardson, "Quality of Service Issues for MPEG-2 Video over ATM," *Proceedings of Internation Broadcasting Convention '96*, Amsterdam, Netherlands, pp. 523-527, September 1996.

[50] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols," ACM Computer Communication Review, April 1997.

[51] J. Rosenberg and H. Schulzrinne, "An RTP Payload FOrmat for Reed-Solomon Codes," IETF Internet Draft, November 1998.

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

[53] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," *RFC 1890*, January 1996.

[54] N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management," *Proceedings IEEE INFOCOM '90*, San Francisco, CA, pp. 124-131, May 1990.

[55] H. Sun, K. Challapali, and J. Zdepski, "Error Concealment in Digital Simulcast AD-HDTV Decoder," *IEEE Transactions on Consumer Electronics*, Volume 38, pp. 108-117, August 1992.

[56] S. Varma, "MPEG-2 over ATM: System Design Issues," *Proceedings of COMPCON '96*, pp. 26-31, February 1996.

[57] O. Verscheure, P. Frossard, and M. Hamdi, "MPEG-2 Video Services Over Packet Networks: Joint Effect of Encoding Rate and Data Loss

on User-Oriented QoS," *Proceedings of NOSSDAV '98*, pp. 257-264, July 1998.

[58] L. Wang, R. S. Ramanujan, J. A. Newhouse, M. Kaddoura, A. Ahamad, K. J. Thurber, and H. J. Siegel, "An Objective Approach to Assessing Relative Perceptual Quality of MPEG-Encoded Video Sequences," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canada, pp. 622-623, June 1997.

[59] Y. Wang and Q-F. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proceedings of the IEEE*, 86(5), pp. 974-997, May 1998.

[60] A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Voran, and S. Wolf, "An Objective Video Quality Assessment System Based on Human Perception," *SPIE Human Vision, Visual Processing, and Digital Display IV*, Vol. 1913, pp. 15-26, February 1993.

[61] S. B. Wicker, "Error Control Systems for Digital Communication and Storage," Prentice-Hall, 1995.

[62] J. Zamora, D. Anastassiou, S.-F. Chang, and K. Shibata, "Subjective Quality of Service Performance of Video-on-Demand under Extreme ATM Impairment Conditions," *Proceedings of AVSPN '97*, pp. 5-10, September 1997.