H.264 Data Partitioned Video Streaming

RICHARD JAMES HAYWOOD

Doctor of Philosophy

ASTON UNIVERSITY

July 2009

Abstract

In recent years the demand for video services streamed over the Internet has grown significantly. This has been exemplified by the rapid growth of dedicated websites such as YouTube and the increase in websites supplementing their written files with video content, such as many news websites like the Times.

Video files are much larger than other types of files typically found on the Internet, such as images or documents. In addition, video streaming imposes strict time constraints on the delivery of data to the client, since the Internet is a best effort network. In this environment, packet loss can occur, especially across wireless links, and due to the timely nature needed for high quality video, retransmission might not be an option. Motivated by the increasing demand and challenges we investigate methods by which the quality of the video can be improved.

We utilise overlay networks that have been created by implementing relay nodes to produce path diversity, and show through analytical and simulation models for which environments path diversity can improve the packet loss probability. We find that in a congested environment the use of path diversity can reduce the packet loss probability, while in both random and burst lossy environments the packet loss probability could be increased with the number of the paths used.

We take the simulation and analytical models further by implementing a real overlay network on top of Planetlab, and show that when the network conditions remain constant the video quality received by the client can be improved. In addition, we show that in the environments where path diversity improves the video quality forward error correction can be used to further enhance the quality.

We then investigate the effect of the recently ratified IEEE 802.11e Wireless LAN standard with quality of service enabled on the video quality received by a wireless client. We find that assigning all the video to a single class outperforms a cross class assignment scheme proposed by other researchers. The issue of virtual contention at the access point is also examined.

We increase the intelligence of our relay nodes and enable them to cache video, in order to maximise the usefulness of these caches. For this purpose, we introduce a measure, called the PSNR profit, and present an optimal caching method for achieving the maximum PSNR profit at the relay nodes where partitioned video contents are stored and provide an enhanced quality for the client. We also show that for the optimised cache the degradation in the video quality received by the client becomes more graceful than the non-optimised system when the network experiences packet loss or is congested.

Acknowledgements

During the completion of this work I have received so much support, ideas and feedback that without these people the work could never have been done. I would specifically like to thank the following people.

My supervisor, Dr Xiaohong Peng, without whom this research would not have been undertaken. During the completion of this work Dr Peng's enthusiasm, support, ideas and direction have greatly influenced the direction of the work undertaken.

University academics including Dr Scot Fowler, Dr David Holding and Professor Keith Blow for their critical comments which have only aided in improving the quality of my work and presentation skills.

My colleagues, Tim, Saty, Guchun and Terence, with whom ideas have been discussed in great detail. I have no doubt that these discussions have greatly aided the work which I have undertaken.

My friends, Neil, Chris, Tim, Mark and many more who have provided time for my brain to unwind and to subconsciously continue working.

Xyratex for their financial support and in particular to Dave Milward and Tim Courtney for their feedback, encouragement and freedom to explore tangents to the original brief.

My family, Margaret, Paul, Matthew, Sophie and most recently Fern. Without their support, feedback and so much more this work would have never been started, let alone finished.

Contents

List of Tables

List of Figures

List of Abbreviations

3GPP	3rd Generation Partnership Project; is a group specifying a standard for third generation mobile phones, page 75
AC	Access Category; different categories allowing service differentiation with 802.11e, page 66
AC_BE	Access Class Best Effort; the best effort traffic class within EDCA, page 79
AC_BK	Access Class Background; the background traffic class within EDCA, page 79
AC_VI	Access Class Video; the traffic class designed for video within EDCA, page 79
AC_VO	Access Class Voice; the traffic class designed for voice within EDCA, page 79
ACA	Autonomous Component Architecture; defines an interface where multiple components are able to communicate, page 91
ACK	Acknowledgement; a packet sent from the destination to the sender to acknowledge the successful receipt of the transmitted packet, page 78
AIFS	Arbitrary Inter Frame Spacing; is the amount of time which the medium must be free before the contention period can start, page 81
AP	Access Point; provides a bridge between a wired and wireless network, page 77
ARQ	Automatic repeat request; a mechanism to request the retransmission of lost or corrupt packets, page 122
AS	Autonomous System; in Internet terms this is a network which is under the control of one organisation, page 68
AVC	Advanced Video Codec; an alternative name for H.264, page 42
B frames	Bidirectionally coded frames; where temporal redundancy from both the preceding and following frames are removed, page 35
B slices	Bi-predicted slices; temporal video compression of a slice using two frames as sources of prediction, page 43
BGP	Border Gateway Protocol; enables the sharing of routing information between routers to enable end-to-end routing of packets, page 68

LIST OF ABBREVIATIONS

Cb	Chroma blue; the blue component of a video frame, page 29
CIF	Common Intermediate Format; a standard frame size of 352 by 288 pixels, page 27
CW_{max}	Congestion Window Maximum; the maximum congestion window size, page 78
CW_{min}	Congestion Window Minimum; the minimum congestion window size, page 79
Cr	Chroma red; the red component of a video frame, page 29
CSMA/CA	Carrier sense multiple access with collision avoidance; a station wishing to transmit must sense the channel first if another station is transmitting then it must wait before transmitting, page 78
CTS	Clear to send; used in conjunction with RTS to enable a client to start transmission, page 132
DARPA	Defence Advanced Research Projects Agency; the US research agency from whom funding started the development of the Internet, page 67
DCF	Distributed Coordination Function; an IEEE 802.11 distributed method to regulate access to the channel, page 77
DIFS	DCF Interframe Space; the duration of time which the station must sense the medium before attempting to transmit, page 78
DS	Differentiated Services; is an 8 bit field replacing type of service within the IP header, page 66
DSCP	Differentiated Services Codepoint; is a 6 bit field within the DS filed to indicate the importance of packets contents, page 66
ECN	Explicit Congestion Notification; is a 2 bit field to enable routers to indicate to end hosts the current network state, page 66
EDCA	Enhanced Distributed Channel Access; provides class differentiated channel access in IEEE 802.11e, page 79
EI	Enhancement Independent; a frame type introduced in H.263+ to enable the scalability of I frames, page 39
EP	Enhancement Predicted; a frame type introduced in H.263+ to enable the scalability of P frames, page 39
FEC	Forward Error Correction; the ability to recover from a certain level of packet loss, page 98
FIFO	First In First Out; a queueing technique where the items taken from the queue is the same order that they arrive, page 87
fps	Frames per second; the rate at which frames are played, page 41

LIST OF ABBREVIATIONS

GOP	Group of Pictures; a contiguity sequence of frames starting with an intra coded frame until the frame preceding the next intra coded frame, page 34
HCCA	HCF Controlled Channel Access; provides class differentiated channel access in IEEE 802.11e, page 79
HCF	Hybrid Coordination Function; extends IEEE 802.11 with the addition of two channel access methods EDCA and HCCA, page 79
HDD	Hard Disk Drive; a drive used for the storage of data within a PC, page 162
HSDPA	High-Speed Downlink Packet Access; a third generation mobile telephone data access technology, page 75
HTTP	Hyper Text Transport Protocol; an application layer protocol within the TCP/IP stack for the transport of data files, page 48
HTTPS	Hyper Text Transport Protocol Secure; an application layer protocol within the TCP/IP stack for the transport of data files over a secure connection, page 48
I frame	Intra frame coding; spatial video compression of a frame, page 30
I slices	Intra slices; spatial video compression of a slice, page 43
IDR	Instantaneous Decoding Refresh; a frame containing only I slices, page 43
IEEE	Institute of Electrical and Electronics Engineers; an international organisation for the advancement of technology in the field of electricity, page 75
IGMP	Internet Group Management Protocol; an IP tool used to managed multicast groups, page 51
IntServ	Integrated services; is an architecture which permits QoS on an IP network, page 65
IP	Internet Protocol; provides a uniform naming standard to enable communication between networks of different types, page 19
IPv4	Internet Protocol version 4; the currently widest deployed version of IP, page 65
ISP	Internet Service Provider; a company which sells connectivity to its network which then connects to other ASs, page 68
ITU	International Telecommunications Union; an organisation established to regulate and standardise audio and video communications, page 28
JPEG	Joint Photographic Experts Group; who develop standards for image compression, page 30
LAN	Local Area Network; an area covering a small physical area, page 76
LFU	Least Frequently Used; a cache replacement algorithm which replaces the least frequently accessed items stored when space is required to store a new item, page 155

LRU	Least Recently Used; a cache replacement algorithm which replaces the least recently accessed items stored when space is required to store a new item, page 155
MAC	Medium Access Control; is a sub layer within the Data Link Layer of the TCP/IP stack and provides addressing and channel access, page 75
MAN	Metropolitan Area Network; is designed to connect a number of LANs to a WAN, page 76
MDS	Maximum Distance Separable; are a set of FEC codes which are able to correct the maximum number of errors, page 121
MPEG	Motion Picture Expert Group; an organisation established to standardise audio and video transmission and compression, page 28
MSE	Mean Square Error; the summation of the square of the pixel differences, page 55
MTU	Maximum Transmission Unit; is the largest packet which can be transmitted without fragmentation, page 100
NAL	Network abstraction layer; the layer of H.264 which passes NALUs to the video transport layer, page 42
NALU	Network abstraction layer units; are generated by the NAL layer and passed on to the video transport layer, page 46
NEGSOB	Negative Sobel Difference; a means to measure the reproduction quality of a frame specifically looking for artefacts, page 55
NRI	Network Reference Indicator; a two bit field within the NAL unit type octet which indicates whether the content of the packet is used as a reference by other slices, page 49
NS-2	Network Simulator version 2; a packet layer network simulator, page 90
P frame	Inter/predicted frame coding; temporal video compression of a frame using a single frame as a source of prediction, page 31
P slices	Intra/predicted slices; temporal video compression of a slice using a single frame as a source of prediction, page 43
PCF	Point Coordination Function; an IEEE 802.11 centralised method to regulate access to the channel, page 77
PEVQ	Perceptual Evaluation of Video Quality; a means to measure the reproduction quality of a frame in a similar way to MOS, page 55
PoP	Point of Presence; a nodes connection to the network, page 67
PSNR	Peak Signal to Noise Ratio; the de-facto measurement of the quality of a resulting frame compared to the original, page 55

QCIF	Quarter Common Intermediate Format; a standard frame size of 176 by 144 pixels, page 29
QoS	Quality of Service; covers a set of factors which effect the quality of the experience received by a user, page 63
RPC	Remote Procedure Calls; enables applications to call software methods which are executed in a remote application, page 47
RS	Reed-Solomon; a set of maximum distance separable FEC codes, page 121
RSVP	Resource Reservation Protocol; a transport layer protocol to reserve network resources, page 65
RTMP	Real Time Messaging Protocol; developed to transport video, audio, remote procedure calls (RPC) and arbitrary data between Adobe Flash player and server, page 47
RTP	Real-Time Transport Protocol; an application layer protocol within the TCP/IP stack for the transport of video, page 46
RTS	Request to send; all clients interested in transmitting data makes this interest know to the access point, page 132
RTT	Round Trip Time; is the duration of time required for a response to be received by a source after sending a message to a destination, page 72
SBP	Scalable baseline profile; a profile to enable scalable video coding by extending a restricted baseline profile from the H.264 standard, page 44
SHIP	Scalable high intra profile; a profile to enable scalable video coding by extending the high profile of the H.264 standard but with restrictions to only intra coded slices, page 44
SHP	Scalable high profile; a profile to enable scalable video coding by extending the high profile of the H.264 standard, page 44
SI slices	Switching I; is a slice defined within the H.264 standard which uses spatial compression to enable the switching between different streams, page 43
SIFS	Short Interframe Space; the duration of time after the medium has become free after which an ACK can be transmitted, page 78
SNR	Signal to noise ratio; a measure of the useful signal against the unwanted noise, page 38
SP slices	Switching P; is a slice defined within the H.264 standard which uses prediction between frames to enable the switching between different streams, page 43
SSIM	Structural Similarity; a pixel by pixel comparison of two frames to identify the quality of the encoded or transported against the uncompressed original, page 56

LIST OF ABBREVIATIONS

SVC	Scalable video coding; a layered video coding technique within the H.264 standard, page 44
TCL	Tool Command Language; a scripting language , page 90
TCP/IP	Transmission Control Protocol/Internet Protocol; is the five layer Internet model shown in Figure 3.1, page 27
ToS	Type of Service; is an 8 bit field in the original IP header design, page 66
TxOP	Transmission Opportunities; allows multiple packets to be sent consecutively without re-contending for the media, page 82
VCL	Video coding layer; the compression layer of H.264, page 42
VoIP	Voice over Internet Protocol; is an umbrella term for the streaming of audio over an IP network, page 68
WAN	Wide Area Network; is a network which covers a broad area with the most well known being the Internet, page 76
WiMax	3rd Generation Partnership Project; is a group specifying a standard for third generation mobile phones, page 76
WLAN	Wireless Local Area Networks; allowing access to a local area network without wires, page 75
WMAN	Wireless Metropolitan Area Network; is a wireless equivalent of a traditional MAN, page 76
WWW	World Wide Web; a collection of application layer protocol to enable browsing and emailling, page 48
Y	Luma; the brightness component of a video frame, page 29

Chapter 1

Introduction

In recent years we have seen the explosive growth of services such as YouTube [8, 9] and more recently the BBC iPlayer [10]. These services provide users with access to a catalogue of video programs over the Internet [11]. Recently, the competition faced by video on demand providers has been increased by services being offered by social networking sites including Facebook [12, 13] and Myspace [14], along with websites choosing to supplement their written online content with video, such as The Times [15] or CNET [16].

Services provided by the likes of YouTube, Facebook and The Times are non-real time video streaming, where the video content is captured, stored, transcoded and then made available for the public to view. For existing broadcasters, such as ITV, BBC and Channel 4, these services will have to be made available online to maintain their current market share. Due to this these, broadcasters are starting to venture into the online video market. Among them, the BBC and ITV are the only terrestrial broadcasters who stream their channels online in the UK.

If current trends for online and on demand video continue then over the next six years there will be a reduction in the viewing of traditional broadcast television by 15% [17]. This will result in a reduction of advertising revenue, in the region of 10%. Should online services not be expanded then illegal activities will fill the gap, which will result in no additional revenue for broadcasters. The broadcasters who develop an online approach are likely to share the predicted £500m advertising revenue [17].

Existing broadcasters currently have the option to stream video over terrestrial, fibre optic or satellite platforms. To enable the delivery of video over the Internet a number of problems need to be solved before video streaming of an acceptable quality can be provided to a user.

1.1 Problems and investigation

There are a number of technical issues surrounding high quality live video streaming over the Internet. Firstly, the video to be streamed must be compressed to a suitable data rate. The choice of this data rate in itself is not easy. If the data rate is too high then the user is unlikely to receive large amounts of the video stream, either resulting in pauses in playback or errors being displayed to the client after the stream has been decoded. If the encoded video rate is too low then the quality of the stream received by the client will be of poor quality, due to compression loss.

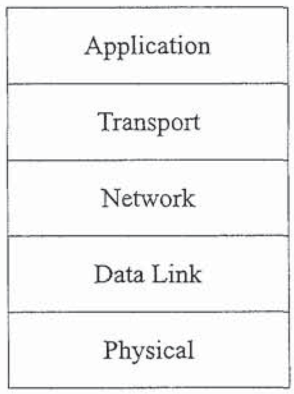


Figure 1.1: TCP/IP Stack

Secondly, the compressed video stream then needs to be transported to the client. When streamed over networks running the Internet Protocol (IP) (shown in Figure 1.1) the video stream is split into a number of packets. Each packet is sent from the server, also termed the source, to the client, also termed the destination, by putting the packets onto the network. Once on the network, the packets will be routed towards their destination by packet switching routers. The majority of packets transmitted are on packet switching networks which are best effort and as such there is no guarantee that any packets put onto the network will arrive at its destination, thus a proportion of the transmitted packets will be lost.

The loss of packets will affect the video quality received by the client. Since the majority of video compression is achieved by removing both spatial and temporal redundancy, when a packet loss occurs it is likely that its effect will remain viewable to the client until the complete screen is refreshed. This generally only takes place every few seconds. If only spatial redundancy were reduced during compression then the error would be displayed on the screen for only a single frame per packet loss, but this requires a much higher bandwidth for the encoded video.

There are a number of techniques which have been proposed to improve the Quality of Service (QoS) which packets receive while being transported across the network. These include the allocation of bandwidth for streams and class based QoS schemes. The problems with allocating bandwidth to streams is that the infrastructure required to provide such a service does not scale to Internet size topologies. The scalability issue comes about because the routers must maintain details about each stream and secondly that with allocating bandwidth it can not be used by another user even if the user to which it is allocated does not require it at that point in time. The class based QoS scheme has been developed as an alternative to the bandwidth allocating scheme and is designed to differentiate based on the importances of packets. Class based QoS schemes are able to scale to Internet size topologies, however, they still provide no guarantees that any packets will be successfully delivered to their destination. Class based QoS requires alteration to routers within the network and these changes are slow and require large amounts of investment.

An alternative to bandwidth allocation and class based QoS schemes is path diversity, this centres around the fact that between any two places on the network there are multiple different routes which a packet can take. Traditionally when multiple packets are sent between the same combination of source and destination, all these packets take the same route. This is because the network layer, which for the Internet uses the IP, is responsible for packet routing and is not concerned about congestion, just connectivity. The problems which result from congestion are transportation layer issues, shown within the TCP/IP stack in Figure 1.1.

There are a number of ways to utilise path diversity, these include source routing, router alteration, multihoming and overlay networks. Of these techniques source routing and router alteration are not generally available options, for reasons explained later. Additionally Path Diversity, enabled by overlay networks, has not been investigated thoroughly with regards to providing video streaming, and the technique poses potential benefits which could result in an improvement in the video quality.

Other researchers have investigated using forward error correction (FEC) to improve the quality of the video received by a client. When a maximum distance separable (MDS) FEC code is used, at the packet level, k packets are put into a group, $n-k$ redundant packets are then added to the group. As long as any k out of n packets are received then the original k packets can be recovered. FEC produces the best effect when errors are dispersed equally across all groups since this reduces the likelihood that k packets will not be received successfully. Path diversity is able to disperse errors within the stream through interleaving. With the dispersal of errors we investigate if additional benefits could be gained when path diversity and FEC are combined.

The last link for clients is increasingly becoming a wireless link. Wireless links experience a higher packet loss rate compared to a like for like wired link. When a video stream is partitioned each of these is not of equal importance, as such if there were a way for the most important partitions of the video to be received with a higher likelihood of being error free than the lesser important packets, even though the packet loss rate for the channel is unchanged, the experience received by the user is improved. The recent IEEE 802.11e standard has facilitated Quality of Service (QoS) within IEEE 802.11 wireless networks, showing potential to improve the QoS received by the client without changing the wireless environment.

Since a significant proportion of the current videos available via the Internet are on-demand we investigate video caching for non real time video streaming. Caching has been used a lot for web content because it provides a number of advantages, these include reducing the number of packet losses as well as reducing the load on a central server and on the network paths over which the data would have travelled. The problem is that video files are far larger than files of other types, as such the number of files which can be stored in a cache will be far fewer than for web caching. To overcome the problem of storing a small number of video we investigate storing only the most important portions of the video stream. This is achieved through intelligent caching which evaluates the importance of each component and only storing the most important ones. The target for this is to provide the client with the best possible video quality both when packet loss occurs and when the original server becomes overloaded. To enable us to measure the importance of each partition we propose a measure, termed PSNR profit, which is calculated by removing select data from the video stream and then comparing this to the original video. The summation of the differences is then used to indicate what the effect of the loss of certain data causes.

1.2 Aims and Objectives

The objective of this work is to improve the quality of the video received by the client when streaming video over packet switching best effort IP networks. We aim to achieve this in two ways; firstly though the investigation of the packet loss resulting from the use of an overlay network and secondly by reducing the visible effect by increasing the likelihood of receiving the most important parts of a video stream. We investigate this with regard to video streaming over both wired and wireless networks with an additional investigation of intelligent video caching. We investigate the effect of intelligent video caching in a lossy environment to identify if the caching of the most important partitions improves the video quality received by the client. In the wireless environment we investigated the effect of using existing QoS parameters which were introduced with IEEE 802.11e for the streaming of video over wireless networks. We aim to investigate if we are able to improve the video received by the client without changing the wireless environment.

1.3 Novelty

This thesis differs from other people's work in a number of ways. Firstly we set out to investigate the effects of using more than two paths in an overlay network, which is the number generally investigated in other peoples work. Through the investigation of more than two paths we aim to see if further improvements in the packet loss rate can be achieved when using path diversity. This investigation is initially conducted through the use of a simulator which is then supported by the presentation of an analytical model. Through the use of the analytical models it is possible to set the parameters for random, burst and congestion losses to see the effect which path diversity would have on the number of packet losses. This work is then taken further by investigating the combing of path diversity with video to see the effect it has on the quality of the video received by the client. In previous works any investigation into video quality has been done with only a single additional path.

Secondly, we perform real experiments using Planet lab to investigate path diversity in the real world. From this we see how path diversity affects packet loss patterns and then we see how this would affect video quality. We then looked at additional benefits offered by path diversity combined with FEC when streaming video. In other works the use of path diversity and FEC have been considered, however, we develop this by applying it to a video environment and, again, apply this to more than two paths.

Thirdly, since an increasing number of clients are connecting to the Internet using a wireless last link, we investigated QoS issues in wireless links for video streaming. In this work we do not aim to reduce the number of packets lost, instead we investigate if it is possible to improve the quality of the video received by the client if lesser important packets are more likely to be dropped than higher importance packets. This is achieved using the recently proposed IEEE 802.11e class based QoS standard. We investigated a previously proposed cross class partition assignment scheme for video streaming over wireless networks using a real testbed. Previously the proposal has only been investigated within the confines of a simulator.

Fourthly, we presented a method to calculate the benefits which are received from the reception of different partition and frame types resulting from compression. Being able to estimate the importance of a H.264 data partitioned unit within a video stream enables more efficient use of techniques such as unequal loss protection or storage.

Finally, we present a method to optimally select which partitions to cache using the previously proposed measure. We present for the first time that H.264 data partitions are used as part of an optimisation techniques used to select which data to store at a cache. Through the storage of the most important parts of the video, clients are able to play back a video with a higher quality when the network experiences packet loss or the original server suffers from congestion.

1.4 Thesis Structure

This thesis is broken up into seven chapters. Chapter 2 discusses the requirements to be able to stream a video over an IP network. We present an overview of the significant developments in video coding from H.261 up to the current state of the art H.264 codec. Following which it goes on to discuss the transportation of video across the Internet. This discussion includes reliable delivery of video using the Transmission Control Protocol (TCP) and unreliable delivery using the User Datagram Protocol (UDP). In addition to a discussion of transport protocols, unicast and multicast are also considered. We then present the most commonly used application layer protocols used for video streaming, these are Hypertext Transfer Protocol (HTTP) and Real Time Messaging Protocol (RTMP) for transportation over TCP and Real-time Transport Protocol (RTP) over UDP. Finally, we present two methods for the calculation of objective video quality. The first method is Peak Signal to Noise Ratio (PSNR), which is an objective frame comparison method which is applied on a pixel by pixel basis, in academe this is the most common objective frame comparison method used. In addition to the pixel by pixel approach we also present the negative sobal difference (NEGSOB) as an alternative approach. NEGSOB differs from PSNR because it provides a measure of quality by first transforming the original and received frames with a sobel edge detection algorithm, it is these resulting frames which are then used to calculate the objective parameter.

In Chapter 3 we look at Quality of Service (QoS) tools which are available within IP networks. This chapter shows that there are limitations to the currently deployed techniques. Firstly the Integrated Service (IntServ) approach does not scale to Internet size topologies. Secondly Differentiated Service (DiffServ) scales to Internet size networks, but provides no guarantee of packet delivery. We then present Path Diversity as an alternative way to improve the QoS of the video stream. Path Diversity has the benefit that it can be selectively deployed on an application by application basis and as such can be operated on the existing Internet. There are four methods to deploy path diversity which are discussed.

Since wireless networks are becoming more common Chapter 3 also includes an overview of QoS tools which are available specifically in wireless networks. These QoS tools have recently been added to IEEE 802.11 networks with the recently ratified IEEE 802.11e standard which we discuss in the chapter.

Since the majority of researchers only look at path diversity from simulations we set out to develop analytical models in Chapter 4 for the packet loss probability of a packet crossing the network. Models were created for environments where losses are caused randomly, in burst and due to congestion. To verify the analytical models we simulate networks and compare the results against the analytical model results. Since the Internet rarely experiences a single type of packet loss we present a combined loss model, which is shown in comparison with the simulation results.

Chapter 4 extends the analytical and simulation models through the use of PlanetLab onto which an overlay network is constructed and experiments are conducted. The construction of an overlay network is achieved by distributing packet relay software onto selected nodes, which forward the packets it receives towards a destination.

In Chapter 5 we investigated the effect of the packet losses on the quality of the video stream. Initially the video is examined in the environment experienced by the PlanetLab overlay network and we show how network characteristics affect the video quality. The work is taken further through the investigation into the improvements offered by the use of FEC.

In addition to the wired work, in Chapter 5 we also investigate the possibilities of using an IEEE 802.11e enabled network to stream class assigned video. Using the assignment already proposed by existing research we take the work which they have conducted through simulation and test it using an IEEE 802.11e test bed.

In Chapter 6 we increase the intelligence of the relay nodes by enabling these nodes to cache video. We show how different components of a video stream are more important than others. To decide which components to store at a cache and to maximise the benefits which are produced through the caching of these components we need to know what benefits their storage offers. To address this problem we present a method to calculate this importance, termed PSNR profit, this is then used to optimise the video components stored at a cache.

The video caching proposed in Chapter 6 is first applied to the case where all video is of equal popularity. We show the effect of the optimised caching compared to the unoptimised caching on the PSNR profit stored at the server. We show how, in a lossy environment, caching the most important video partitions improves the quality of the video received by the client. The single cache is then expanded to a caching network of nodes which ensures that a complete copy of the video remains available across the whole network. Finally the caching is adapted to the case where the popularity of the video is unequal and we show how this affects the proportions of each component stored at the cache.

Finally in Chapter 7 we present an overview of the work completed along with a summary of the conclusions which have been drawn. In this chapter we also highlight the five contributions which this thesis has made to the advancement of knowledge within our field.

Chapter 2

Video Streaming Technologies

The purpose of video streaming is to transport video from the source to the destination in a timely manner for playback by the software running at the destination. Within this thesis we restrict our discussion to video streaming over IP networks.

One of the first problems with video streaming is the high bandwidth which raw video requires. We can represent each pixel of a frame with three byte values, one each for red, green and blue. The raw data rate of a Common Intermediate Format (CIF) size video, which has a frame size of 352 pixels by 288 pixels, and a frame rate of 24 frames per second, is 6.96 megabytes per second (55.69 megabits per second). When this is extended to high definition video with a frame size of 1920 by 1080 pixels the raw data rate rises to 142.38 megabytes per second (1.11 gigabits per second) [18]. In the majority of instances streaming data at this rate is unachievable with currently available technology where the highest home connections are about 20 megabits per second. If we store an average two hour long video the raw data would take up 0.98 terabytes, which exceeds the capacity of most currently available hard drives which are typically around the 750GB. A number of standards have been issued to enable the compression of raw video. The major developments in video compression standards are presented in Section 2.1.

Once a video stream is compressed we need to transport the video to the client. There are two major transport protocols within the Transmission Control Protocol/Internet Protocol (TCP/IP) stack. A discussion of transport protocols and the effect they have on video streaming is included in Section 2.2.

The video decoder is required to produce complete frames for playback, since when a video stream experiences packet loss the decoder must fill in the missing areas. There are a number of different methods which can be used to fill in areas of the video. The selection between these methods is dependent on the software used by the client. In Section 2.3 we present the most common algorithms used to fill in the missing data of a frame.

Once the complete frames are available it is possible to compare them against the frames which were sent from the video server for the purpose of video quality assessment. There are two methods to measure the quality of the video received by the client, subjective and objective quality measurements. In the subjective measurement a large number of users are asked to rate the quality of the video they have watched. The subjective quality measurement is very labour intensive and thus the use of objective quality measurement is more common; since they can be processed automatically by computers. In Section 2.3 two objective quality measures are presented.

2.1 Video Compression

To reduce the data rate required by the network many video compression standards have been developed from both the International Telecommunications Union (ITU) and the Motion Picture Expert Group (MPEG). In this section we overview some of the milestones in video coding, ending with the current state of the art video codec.

2.1.1 H.261

H.261 is a source coding standard developed by the ITU [19] and contains the fundamental principals on which the majority of subsequent video coding standards are based. The standard is designed for ISDN lines and works with constant data rates which are multiples of 64kbps. H.261 and the video codes that follow are lossy codecs, i.e. the compressed video does not decode identically to the original.

The basic design of the video encoder is shown in Figure 2.1. The video standard only defines the bit stream format which dictates the basic decoder design. The advantage of standardising the bit stream and no other components is that this enables the encoder and decoder to be optimised for their particular use. In addition to the encoding and decoding of the video, there is also the option for pre-encoding and post-decoding processes to be used. One example of these might include deblocking filters for the removal of compression artefacts after decoding.

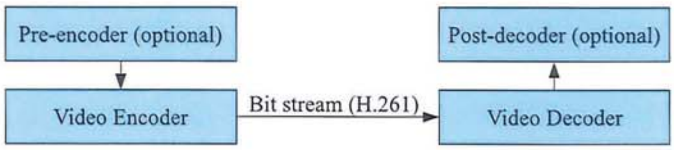


Figure 2.1: Video encoding block diagram

H.261 supports two different sized frames, CIF (352 by 288 pixels) and QCIF (176 by 144 pixels), which is a quarter of the size of CIF. Each CIF frame is split into two columns and six rows which totals 12 groups of blocks. In a QCIF frame, there are only three groups of blocks but each group is the same size as the groups in a CIF frame. Each group of blocks contains 33 macroblocks in an 11 by 3 formation.

Each H.261 macroblock is encoded using the YCbCr colour space. YCbCr splits an image into three components, the luma (Y) and two colour difference components for blue (Cb) and red (Cr).

The human eye is more sensitive to changes in luma value than it is to chroma values. Through the separation of the chroma and luma components of the video it is possible to reduce the number of chroma components, with only a limited reduction in the perceived video quality. H.261 takes advantage of the split luma and chroma by separating a picture down into a number of macroblocks represented by a 16 by 16 array of samples for the luma and two 8 by 8 arrays for Cb and Cr. These details can be represented by the chroma subsampling ratio of 4:2:0, as shown in Figure 2.2. This results in four times more luminance information than colour information being transmitted.

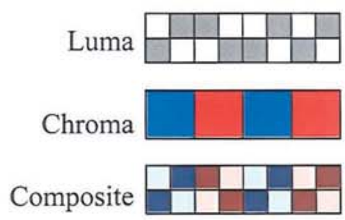


Figure 2.2: Subsampling at 4:2:0

H.261 includes two methods of frame compression, intra and inter. In Subsection 2.1.1.1

an overview of the encoding of intra frames is presented, and then in Subection 2.1.1.2 the differences between inter and intra coded frames are explained.

2.1.1.1 Intra Frame Coding

Intra frame (I frame) coding compresses frames through spatial compression. In this section we provide an overview of the steps involved in the compression of a frame using intra coding.

Discrete cosine transforms Firstly a discrete cosine transform is used to transform an image from the spatial domain into the spatial frequency domain. The discrete cosine transform takes an n by n matrix, f where $f(i,j)$ is the intensity of the pixel in the i^{th} column and j^{th} row. In the case of H.261 $n = 8$.

$$f = \begin{pmatrix} f(0,0) & \dots & f(n-1,0) \\ \vdots & & \vdots \\ f(0,n-1) & \dots & f(n-1,n-1) \end{pmatrix} \tag{2.1}$$

The resulting matrix, F, has components $F(u,v)$ which refer to the amplitude of the frequencies present within the original 8 by 8 matrix. The lowest frequency is in the $F(0,0)$ position, termed the DC value, and the horizontal and vertical frequencies increase as u and v rise respectively.

$$F = \begin{pmatrix} F(0,0) & \dots & F(n-1,0) \\ \vdots & & \vdots \\ F(0,n-1) & \dots & F(n-1,n-1) \end{pmatrix} \tag{2.2}$$

The human eye is more sensitive to lower frequencies and so some of the higher frequencies could be discarded without much noticeable degradation in the perceived quality.

Quantisation Quantisation is implemented by dividing the content of the resulting matrix, F, by a constant value and rounding the result, meaning that there will be a reduction in the quality of the representation but this also means that there will be a reduction in the amount of data transferred. If the reduction in quality cannot be noticed then quantisation does not have a detrimental effect. The alternative to dividing each element of the matrix by the same value is to use a quantisation table, as used in the Joint Photographic Experts Group (JPEG) image coding standard. This means that the same amount of data could be distributed better, or less data could be used to transfer the same information as more important elements of the matrix could be quantised less than less important elements.

Run length encoded Through the use of a discrete cosine transform and quantisation the frame has been compressed. The resulting values are then converted into a stream. The resulting stream generally has a large number of zeros grouped together. Run length encoding is used to represent the sequence using skip, value pairs. The value term is the next non-zero component and skip is the number of zeros preceding it. As the resulting sequence is not of a fixed length an end of block marker is appended. For example the sequence 160,0,1,0,0,1,0,0,1 would be transformed into 160 (1,1) (2,1) (2,1) END.

Huffman coding Huffman coding is a lossless compression algorithm that can be implemented by using a lookup table to encode the data. The conversion from the symbol to a variable length bit sequence is based on the probability that a certain symbol will be sent next. The combination of run length and Huffman coding provides a reduction in the region of two to three times less data than the raw output from the quantiser, all without any additional loss in visual quality [20].

2.1.1.2 Inter Frame Coding

Whereas intra frame coding only compresses spatially, inter frame (P frame) coding is able to use the temporal redundancy exhibited in video. There is a significant amount of temporal redundancy and the removal of this can reduce the data rate required to stream a video. Figures 2.3a and 2.3b show two consecutive frames from a sample video sequence. Figure 2.3c shows the difference between Figure 2.3a and 2.3b from which we can see that the majority of the frame remains unchanged.

The reduction in the data rate is at the cost of increased complexity at both the encoder and decoder. For the video to be played back at the client with only the affects of compression there must be no loss of data in any of the frames received, both for inter as well as intra frames, as illustrated in Figure 2.4.

(a) Sample frame 1

(b) Sample frame 2

(c) Difference between frames 1 and 2 shown in white

Figure 2.3: Illustration of the similarity between two consecutive frames

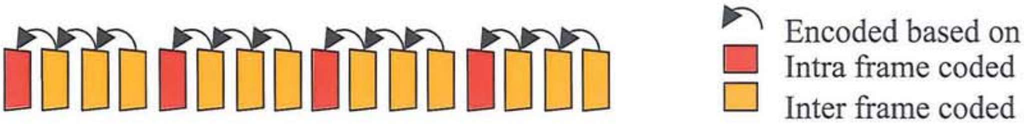


Figure 2.4: P and I frames and their relationships

The use of inter frame coding imposes additional processing and storage demands on both the encoder and decoder. To calculate motion vectors the encoder needs to store a copy of the preceding frame in order to encode the predicted frame from it. To match the decoder, the encoder must use a decoded version of the frame. This is because the original raw frame will not be available at the decoder.

Block-based motion compensated prediction When a block in a frame is compared to the corresponding block in the previous frame it is unlikely that the blocks are identical. Figure 2.5 shows two sample consecutive frames within a video. When we compare two corresponding blocks from the frames we can find that there is no repetition between them, as shown in Figure 2.6.

Instead of doing a direct block to block comparison, we can search within the frame for a block sized area that matches closely the future block. To find the closest matching area to use as the source of prediction of the macroblock, a search is conducted to find an area that has either the lowest mean absolute distance, the lowest mean squared error or the lowest sum of absolute differences, the choice of which is encoder specific.

To find the closest matching block size area within a CIF frame would require more

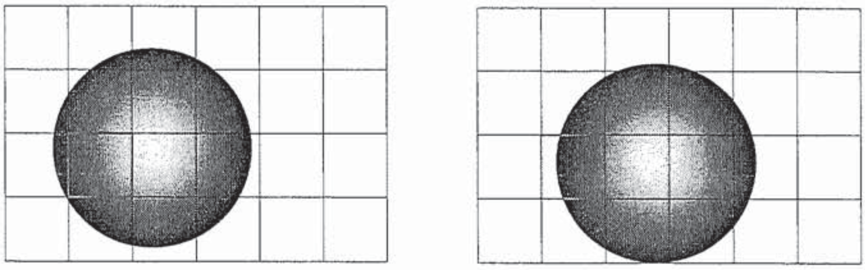


Figure 2.5: Two consecutive frames of a sample video sequence

Figure 2.6: The area highlighted in Figure 2.5 shown side by side to illustrate the lack of redundancy when two blocks are compared directly to each other

than 91 thousand block size comparisons. Since H.261 only defines the bit stream and the basic decoder, it is possible for different H.261 encoders to optimise the search algorithm for a particular task, or even ignore it and transmit a $0, 0$ motion vector for the block, but this causes an increase in the video bandwidth. Potential search techniques include two dimensional logarithmic search, hierarchical search or a finite area search, as shown in Figure 2.7. These different search techniques each add a different level of computational complexity into the encoder.

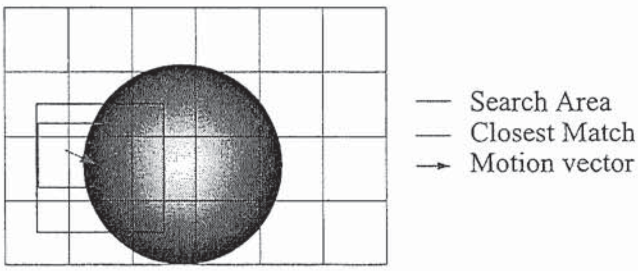
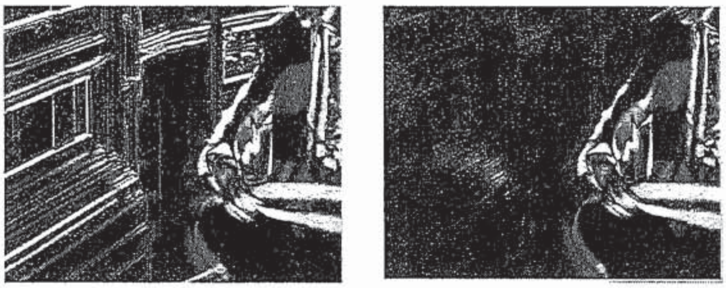


Figure 2.7: Motion prediction vector illustration

Figure 2.8b shows the difference between the frames shown in Figures 2.3a and 2.3b after using motion prediction. Figure 2.8a shows the difference between Figures 2.3a and 2.3b without motion prediction. From these we can see that the differences between the two frames using motion prediction is far smaller than when frames are compared directly.



(a) Difference between frames 1 and 2, from Figure 2.3, shown in white

(b) Difference between frames 1 and 2, from Figure 2.3, with motion compensation

Figure 2.8: Illustration of the impact of motion compensation

2.1.1.3 Group of Pictures

The reduction in temporal dependency comes at the cost of error propagation. If data within the bit stream is lost or corrupted then an error will be displayed on the screen. When future P frames are decoded, the error will be copied forward. To remove the temporal dependency intra coded frames can be used periodically, a group of pictures (GOPs) starts with an intra coded frame and contains all the following frames up to and excluding the next intra coded frame. The problem of error propagation can be reduced by increasing the frequency of I frames. However, since P frames have a higher compression efficiency this means that a higher bandwidth is required when the frame rate remains constant. Figure 2.4 shows the prediction used for inter frames, within this figure there are four GOPs shown.

In addition to the reduction in the propagation of errors within the video stream, GOPs also provide random access into the video stream to facilitate such features as fast forwarding and rewinding. If a video sequence comprised of an I frame followed by a number of P frames, then to decode any frame would require decoding all of the previous frames. With the use of GOPs it is possible to decode from a larger number of points within the video where no previous frames are required, enabling random access within the video which would give the user the facility to fast forward and rewind. Reducing the size of the GOP would increase the potential entry points into the stream, but because I frames cannot be compressed as highly as P frames, the data rate required increases as the size of the GOP falls.

2.1.2 MPEG-1

The MPEG-1 standard is developed by MPEG to compress video and audio to a rate of 1.5 megabits per second at a quality equivalent to VHS video [21]. The standard includes methods for the synchronisation and multiplexing of video and audio streams, compression for non-interlaced video streams and audio as well as reference software.

The majority of the compression process is very similar to that of H.261, but there are some changes which improve MPEG-1. In H.261 there are two types of frame, I and P. MPEG-1 notably adds a third type of frame, bidirectional (B frames). In H.261 P frames are predicted from either I or P frames, which is still the case for MPEG-1.

2.1.2.1 Bidirectional Frame Coding

Bidirectional frames (B frames) are predicted from the next I or P frame and the last I or P frame. Figure 2.9 shows one complete GOP starting with an I frame, and one partial GOP (containing the last five frames) with the addition of bidirectional frames.

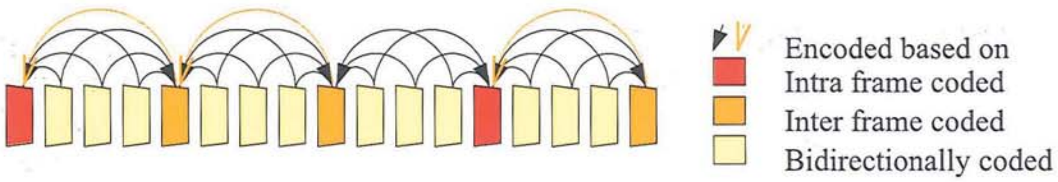


Figure 2.9: MPEG-1 frames and their dependencies

To encode B frames, the future frame, both intra and inter, must be encoded and decoded before the B frame is encoded or decoded respectively. Storing multiple reference frames increases the buffer space required and out of sequence encoding and decoding results in a longer processing duration. As a consequence the frames are not streamed in the order in which they are played out.

The encoding efficiency of B frames is greater than that of P or I frames with a typical B frame being two fifths the size of a P frame, which in turn is a third the size of an I frame. The amount of data saved by using B frames does come at a price of computing twice as many motion vectors as inter coded frames, once for the preceding frame and once for the following frame.

B frames are never used as a reference frame for any other frame. As such if the frame gets corrupted then the corruption will only be displayed in a single frame. Since these frames are not used for reference, unlike the other frames, they can be viewed as being less important. For example with a GOP of 132 frames, should either a P or I frame be corrupted then artefacts will be present in, on average, 66 frames, whereas artefacts in a B frame will only be present in the single frame. With a frame rate of 24 frames per second, the average duration for the corruption of an I or P frame to be visible for is 2.75 seconds, compared to 0.04 seconds for the loss of a B frame.

2.1.3 H.262 and MPEG-2

The video components of MPEG-2 and H.262 are jointly developed to create a standard designed specifically to take into account the requirements for video broadcast but still based on the MPEG-1 coding method [22]. MPEG-2 introduces the idea of profiles which only refer to the resolution and frame rate, but in later codecs they also provide different tools. The main contribution of the H.262 standard is to introduce ways to encode interlaced video.

MPEG-1 only permits progressive images which, when displayed on the screen, scan from the top of the screen to the bottom. This is different from the scan used in most TV broadcasts which uses interlacing to increase the refresh rate of the screen without any increase in bandwidth. Interlacing scans the screen twice to display a full frame. These two scans are called fields, one field contains the odd lines and the other field contains the even lines. Interlacing produces a number of problems for video compression.

With interlacing, one frame is made up of two fields which are captured at slightly different times. This produces a problem for the encoder as it is highly unlikely that during this interval items in the frame content have remained static. If the two fields are combined to form one frame it would be possible to use these for predicting one frame from the previous one but there are two problems with this.

Firstly, this produces lines through the image, and when passed through MPEG's lossy compression these high frequencies are likely to be far less accurately decoded at the receiver than is required for accurate reconstruction of the two independent fields. However if there is no movement in the frame, this is still a good strategy. Secondly, when movement occurs, combining the two fields fails to produce a high quality output at the receiver. The alternative is to encode the two fields separately and this is achieved by only having lines in the block from one of the fields, as illustrated in Figure 2.10. The decision to encode based on the field or frame is made at the macroblock level. It is the macroblock setting that determines the block arrangement for the entire macroblock.

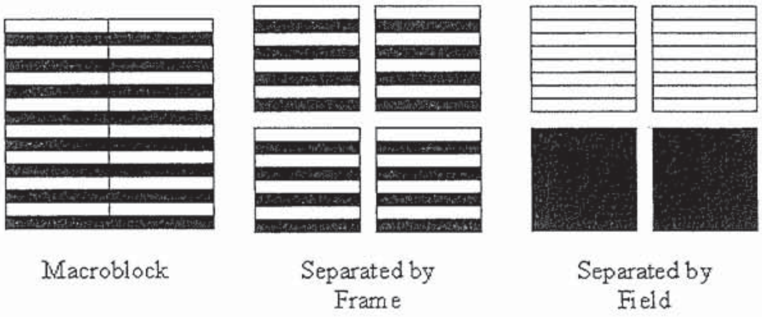


Figure 2.10: Illustration of field separation techniques

To complement the separation of the macroblock depending on the frame or field of the original source, the motion vector is given three modes. Firstly, the frame mode which is the same as the motion vector in MPEG-1 where the current frame is predicted from the previous frame. The second is field mode where, instead of presuming that both fields were taken at the same time, the prediction for the current field is based on the previous field. Finally, there is a mixed mode, where the prediction is based on two macroblocks in two different frames.

2.1.4 H.263

H.263 is developed by the same group as H.261. One of the main differences from previous standards is the introduction of options [23]. The options provided by H.263 can be used when the client supports them and the source wants to use them. The problem with these options being negotiable is that they have to be used on the fly. Alternatively, multiple encoded copies of the same file need to be stored in order to deliver a video which matches the options returned by the client. In a web based environment, this is not such an issue as software for supporting the options that the server wants to deliver can be pushed to the client's browser.

Following the H.263 standard two further versions of the standard are released, entitled H.263+ and H.263++, which provide additional optional encoding tools. The most relevant additional options are presented in Subsections 2.1.4.1 and 2.1.4.2.

2.1.4.1 H.263+

H.263 version 2, also known as H.263+, extends H.263 with the addition of a number of additional optional coding modes and more choices over picture sizes up to 2048 by 1152 pixels [24, 25]. We highlight three options; slice structure mode, signal to noise ratio (SNR) and spatial scalability mode. Additionally, H.263+ introduces Reference Picture Selection Mode, which is further improved in H.263++.

Slice structure mode Instead of using the group of blocks structure, first used in H.261, a slice structure is used. A group of blocks contain the macroblocks in an integer number of rows. The difference between a group of blocks and a slice is that a slice can contain a variable number of macroblocks spanning partial rows, whereas slices do not use any data from surrounding slices, meaning they can be decoded separately.

There are a number of benefits of the uses of slices. Firstly, it enables the encoding of frames to be multi-threaded. Each thread is able to process a slice independently of the compression of any other slice. The use of slices is ideally suited for the environments where the frame is naturally divided into different regions, such as news channels which contain a ticker across the bottom and a main feature panel. Another useful result of slices is when multiple slices are used to compress a frame, then any error that might be created is restricted to a smaller region of the frame.

Scalability modes H.263+ introduces two new frame types called Enhancement Independent (EI) and Enhancement Predicted (EP) frames which are combined with B frames, permitting three scalability modes. Scalability modes allow encoded videos to be split into multiple streams. It is possible that as long as the base layer is received some form of playback will be visible to the client and with each enhancement layer received, the reconstructed scene received by the client will be improved.

Temporal scalability mode Temporal scalability provides a basic frame rate, and with every enhancement layer received the frame rate increases. It is very simple to achieve in most video standards since MPEG-1, through the use of B frames. B frames are not used as a source for the prediction of any other frames and so if they are lost the remaining stream would be able to continue playback. The layering is achieved by grouping B frames into layers, as shown in Figure 2.11.

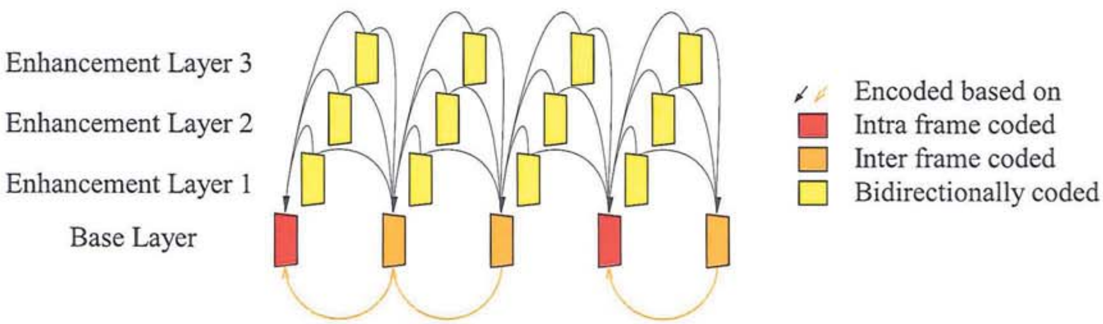


Figure 2.11: Illustration of temporal scalability mode

Spatial scalability mode Spatial Scalability allows for the base stream to cover just the essential parts of the video and additional layers can be used to extend the area up to the full capture area. Figure 2.12 illustrates spatial scalability mode with a base layer and an enhancement layer.

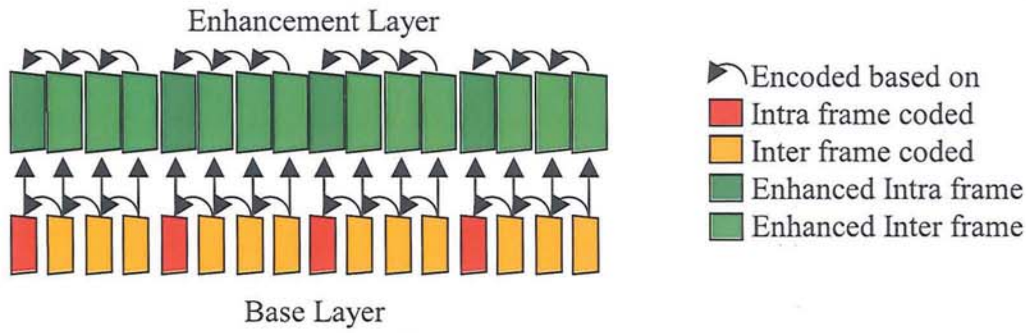


Figure 2.12: Illustration of enhanced layer spatial scalability

Signal to Noise Ratio scalability mode When Signal to Noise Ratio (SNR) scalability is used every layer contains all of the frame area, but captured at a lower resolution. The base layer contains all the motion vectors and the enhancement layers just improve the resolution. This is shown in Figure 2.13. Potentially this is far more useful than spatial scalability as most programs are recorded in such a way that the majority of the screen is used. Therefore it would not be possible to select just part of it for being encoded as the base layer.

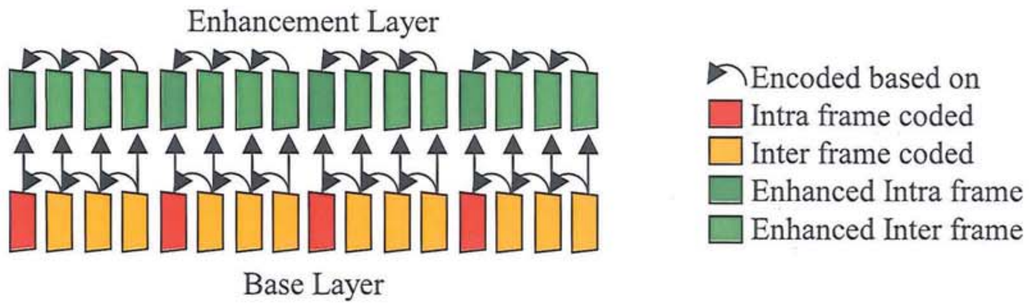


Figure 2.13: Illustration of enhanced layer SNR scalability

2.1.4.2 H.263++

H.263++ provides a further four options to H.263 and H.263+ [26]. An H.263++ decoder is backwards compatible to support H.263 and H.263+ streams. The additional annexes are enhanced reference picture selection mode; data partitioned slice mode; profiles; levels and finally additional supplemental enhancement information.

Enhanced reference picture selection mode H.263++ extends the reference picture selection mode originally provided by H.263+. Enhanced reference picture selection mode provides a bit rate saving of 10-20% when 10-20 reference frames can be used [27]. A simplified example of reference picture selection is shown in Figure 2.14. Reference pictures are selected on a macroblock level and can include a multi-frame motion prediction to increase the coding efficiency. If multiple frame reference pictures are used then this should reduce the possibility that an error occuring at the decoder will be copied onto subsequent frames.

To save space reference picture pruning is used. Each picture stored in memory is partitioned into multiple sub-frames and each of these can be dropped from the buffer when the encoder no longer feels that it will be useful. For the encoder to drop the correct sub-frames would require the ability to look ahead and decide on the most optimal way to use the available storage, but unfortunately this increases the delay between encoding and transmission which would not be suitable in a live streaming situation.

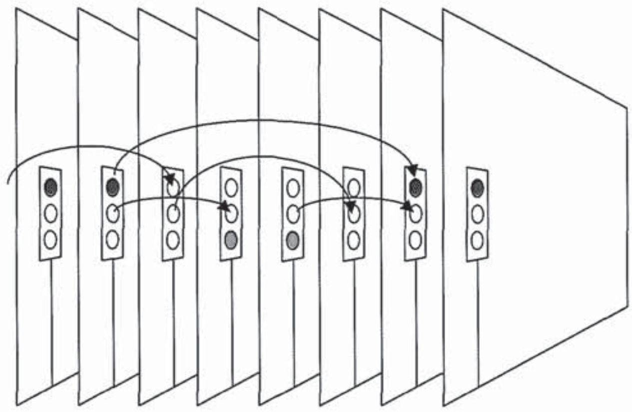


Figure 2.14: Illustration of H.263++ reference picture selection mode

Profiles and levels definition The number of available options for H.263 with the extensions made by H.263+ and H.263++ means that there are over 300 different combinations of options. The aim of profiles is to reduce this to 9 different options.

To support a number of different frame sizes and data rates, levels are also introduced. These levels range from level 10 which supports QCIF and Sub-QCIF at 64Kbps with a rate of 15 frames per second (fps) up to layer 70 which supports custom and standard frame size up to 720x576 at 16Mbps, with a rate of 50fps for frames smaller than 720x576 and a rate of 60fps for frames smaller than 720x480.

2.1.5 H.264

H.264 is a joint partnership, in the same way as H.262 and MPEG-2, between the ITU and MPEG. H.264 is part 10 of the MPEG-4 standard and is also known as the Advanced Video Codec (AVC) [28]. H.264 manages to offer twice the performance compared against the MPEG-4 part 2 video codec [29–32] through a number of changes. These include integer transforms, entropy coding, permitting multiple reference frames, intra prediction, an in-loop de-blocking filter, SP and SI slices and new error resilience tools. This reduction in bit rate does come at a price with H.264 decoders being four times more complex than those for MPEG-2 [33] and two and a half times more complex than a H.263 decoder [34].

Where as previous coding standards aimed to produce an output stream, H.264 splits the encoding from the delivery of the compressed video. The separation of the delivery encapsulation from the video coding provides a number of benefits and simplifications. For example, streaming compressed video over a satellite link would require items such as synchronisation markers, which are not required when transporting the stream over the Internet since these would be provided by another layer of the TCP/IP stack [35]. Figure 2.15 illustrates the split encoder design. Firstly the video coding layer (VCL) compresses the video and, secondly, the network abstraction layer (NAL) encapsulates the video into the required stream format.

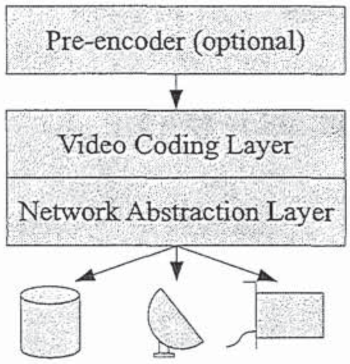


Figure 2.15: Illustration of H.264 codec layers

2.1.5.1 Video Coding Layer

H.264 is a coding standard which, like a number of standards before it, processes macroblocks [31, 36]. Each macroblock consists of four luma blocks and two chroma blocks. The H.264 standard includes the in loop deblocking filter (developed from H.263+) and the use of multiple reference pictures (developed from H.263+ and H.263++). H.264 also makes some additions, with the main ones being intra compression using surrounding previously received macroblocks and the use of an integer transform instead of the discrete cosine transform used previously. The groups of blocks that were present in previous standards have also been replaced by slices (similar to the slice structure mode option in H.263+). A slice is able to contain any number of macroblocks and is decoded in a raster scan order.

Slices Each video frame is encoded into one or more slices. Within a frame each slice can be of a different slice type. The H.264 standard defines five different slice types.

Intra slices (I slices) are encoded using only intra-coded macroblocks. Intra coded macroblocks are not encoded identically to previous standards, but it still holds true that an intra coded slice can be completely decoded without the prediction from other frames.

Inter slices (P slices) can contain inter, intra or skipped macroblocks. H.264 uses lists of reference pictures to allow the encoder to remove even more redundancy by using closer matching frames. When a frame is encoded it is assigned to List 0 as a short term reference picture. When the memory is full the oldest short term reference frame is removed. A short term reference picture can be assigned as a long term reference picture. Long term reference pictures are not removed from memory unless explicitly removed or replaced. On the reception of an Instantaneous Decoding Refresh (IDR) all pictures in memory are marked as "unused for reference".

Bi-predicted slices (B slices) can be predicted from one or two reference pictures, both before or after the current frame in temporal order. Unlike I and P slices, which have a close resemblance to I and P frames from previous encoding standards, B slices are subtly different from B frames. In previous standards, macroblocks of a B frame are predicted from a future and past frame. H.264 B slices are not as restrictive as B frames, allowing the use of a past and future frame, or either two past frames or two future frames. With both past and future frames present in the picture buffer, two lists are kept, termed List 0 and List 1. List 0 is ordered such that past frames are ordered first followed by future frames, whereas List 1 is ordered with future frames first followed by past frames [37].

The standard also details switching P (SP) and switching I (SI) slices, which are coded to enable the switching between video streams. These could be used in an Internet environment where multiple streams are available at different bit rates and the decoder might swap between streams as the available bandwidth changes. It would be possible to switch between streams on the reception of an I slice, but SP slices use prediction and thus require a lower bit rate. SI slices work similarly but are designed to be used at scene changes where there is little prediction possible [29].

Scalable video coding Layered codes provide a mechanism to encode a video stream into a base layer and multiple enhancement layers. The advantage of this is that as long as the base layer is received the client will receive a standard level of service with each additional layer received enhancing the user's experience.

Scalable Video Coding (SVC) is a relatively recent addition to the H.264 standard with layered coding. The development of SVC has created three new H.264 profiles, Scalable Baseline Profile (SBP), Scalable High Profile (SHP) and Scalable High Intra Profile (SHIP). SBP extends a restricted baseline profile whereas both SHP and SHIP extend the high profile.

Within SVC there are three scalable options; spatial, quality (in principle similar to the SNR scalability mode of H.263+) and temporal [38]. SBP supports a restricted spatial scalable coding as well as quality and temporal scalable coding. SHP and SHIP both support all tools specified in the SVC extension. Spatial scalable, quality and temporal scalable coding are supported without any restriction.

Scalable video coding is a layered coding technique. Previously, layered coding had never been widely used. The reason for the limited take up can be put down to a number of reasons, including a significant loss in compression efficiency when spatial or quality scalability is used and a significant increase in decoder complexity [38]. These issues need to be overcome in order for developers to develop their products using these techniques.

SVC is able to provide a coding efficiency similar to that of regular H.264 with only a single layered output. For the same fidelity the use of SVC can require 10% additional bandwidth [38]. Additionally, SVC requires little additional complexity for decoding the stream at the receiver.

Data partitioning Data Partitioning is not a layered codec, however it is able to provide multiple streams of different importances at the same data rate and also with little additional complexity at the decoder. This is archived based on the fact that compressed video data is not equally important.

When an H.264 video is encoded without Data Partitioning [39], each slice of a frame is encoded into one NAL unit. When H.264 encodes frames with data partitioning enabled, up to three NAL units per slice are produced. The three NAL units are named partitions A, B and C, respectively. Figure 2.16 show a possible assignment of encoded slices into partitions.

Partition A contains the most important elements of the slice, including the slice header, macroblock types, quantisation parameters, prediction modes and motion vectors. Partitions B and C contain the residual information for the intra and inter coded macroblocks, respectively. If partition A is lost then partitions B and C cannot be decoded. However, if partition A is received then the quality of the displayed frame will be improved when partition B or C

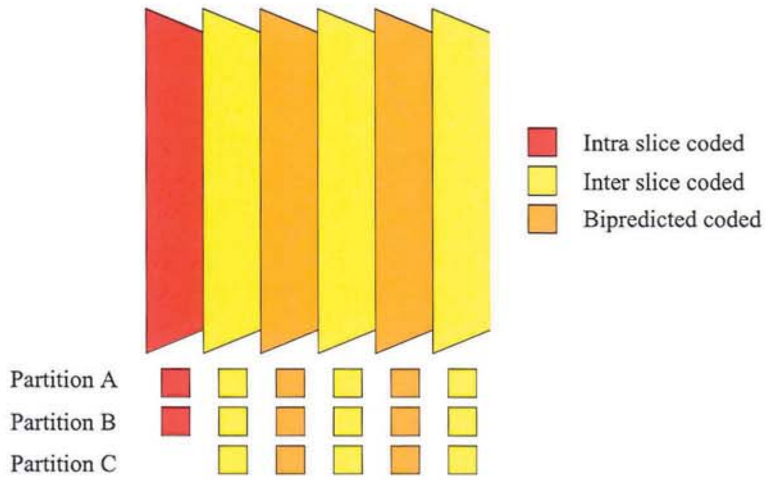


Figure 2.16: H.264 data partitioning example

is received as well.

There are a couple of advantages of data partitioning over SVC. Firstly since the data is only rearranged there is little additional overhead created and with minimal additional decoder complexity [37]. Secondly, the authors of [40] showed, through simulation, that the use of H.264 data partitioning provided a significant improvement (over 10 dB) over systems without data partitioning. The paper went on to compare data partitioning with the multiple description code and found that at bit error rates lower than approximately 4×10^{-4} data partitioning performed better.

2.1.5.2 Network Abstraction Layer

The separation of the NAL enables the video stream to be suitably packaged for whichever medium it is going to be stored or transferred across. The NAL provides an interface between the VCL and a video transport layer (not to be confused with the transport layer in the TCP/IP stack). The video transport layer could transfer the video stream in an MPEG-2 transport stream, an ISO media file format or Real-Time Transport Protocol (RTP) packets for streaming over the Internet [41]. The NAL generates Network Abstraction Layer Units (NALUs) which are then passed to the video transport layer. Within our work we limit our investigation to IP networks.

2.2 Video Transport Over IP Networks

Once compressed, the video data needs to be transported from the source to the destination. Within the TCP/IP stack there are two commonly used transport protocols, TCP and UDP. Within this section we review both TCP and UDP. For each we present a typical application layer protocol used to receive the video and compare them for the purposes of video streaming.

2.2.1 Transmission Control Protocol

TCP provides reliable delivery of packets to the application layer. When packets (called TCP segments) are lost TCP automatically requests a retransmission. In addition, TCP contains a congestion avoidance algorithm. The exact details of the congestion avoidance algorithm forms a large research area in its own right, but is outside the scope of this thesis. However a general understanding is useful to identify its effect on video streaming.

TCP uses acknowledgements to indicate which packet sent from the source has been received at the destination. These are used to influence the size of the congestion window. When packets are successfully received by the destination the source increases the size of the congestion window. However, when a packet is lost the congestion window is reduced. The exact reduction is dependent on the TCP algorithm used. In earlier versions of TCP the loss of a single packet resulted in the congestion window being halved. There are many TCP variants used, this could be Tahoe, Reno, Vegas, BIC, CUBIC or one of many more. Taking CUBIC as an example [42], as it is the current implementation in the Linux Kernel, when a packet is not received by the destination the contention window is reduced by 20% [43]. Figure 2.17 shows the effect on TCP Cubic's throughput with different packet loss rates, where the packets are lost in a uniform random manner.

When a packet loss occurs TCP requests a retransmission. While waiting for the lost packet to be received, any packets received later in the video stream are cached by TCP. This caching can cause pauses since the video client will not receive these packets until the lost packet is received.

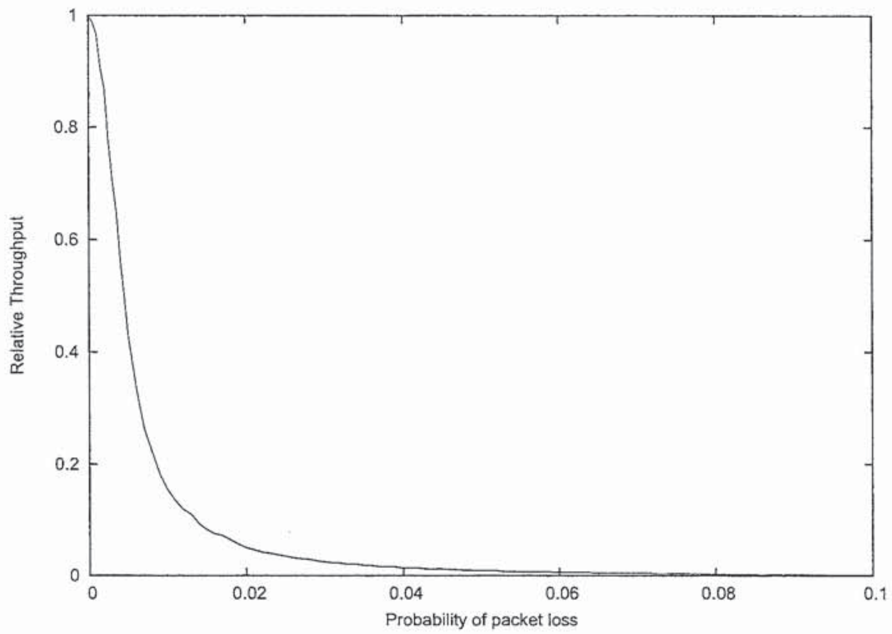


Figure 2.17: The effect on relative throughput of uniform random packet loss [3]

2.2.1.1 Real Time Messaging Protocol

Real Time Messaging Protocol (RTMP) is a protocol developed by Adobe for the transport of data between its server and client software. The RTMP protocol was developed to transport video, audio, remote procedure calls (RPC) and arbitrary data. RTMP is a closed source protocol and as such the exact details of it are unknown.

A number of popular video services use RTMP to some extent, including the BBC's iPlayer and YouTube. Sites like YouTube which play back video in the browser use the Flash player software and this is also the same for the BBC's iPlayer online version. For the offline version of iPlayer the Adobe Air framework is used. Both of these pieces of software use RTMP to transport the video stream.

Due to issues with firewalls restricting access to only HTTP streams, and thus filtering out access to RTMP streams, RTMP provides an option to tunnel RTMP streams over Hyper Text Transport Protocol (HTTP) or the secure variant of HTTP, called Hyper Text Transport Protocol Secure (HTTPS). To the firewall, the stream appears as thought it obeys the HTTP or HTTPS protocol, but after the passing of HTTP headers the stream implements the RTMP protocol. If HTTPS is chosen then the RTMP communication is exchanged securely.

The problem with RTMP is the selection of the rate at which the video is encoded. If this rate is too high then pauses in playback will occur while the client has to buffer more data. When packet loss occurs TCP will automatically ask for retransmission. If the time taken for the server to retransmit the required packets takes longer than the amount of time to play out the data buffered then this will result in a pause in the playback at the client.

2.2.1.2 Hyper Text Transfer Protocol

The hyper text transfer protocol (HTTP) is a protocol designed for the delivery of data files using TCP as its transport protocol. HTTP is used in the delivery of content within the world wide web (WWW). The advantage HTTP has over RTMP is that it is open source and has been very widely used. Therefore, people are more familiar with dealing with HTTP traffic than with RTMP, and open source software is widely available to implement the protocol. The advantage of open source software is that it can be customised for the particular requirements of video streaming. There are two ways in which HTTP is used to transfer a video, termed progressive and delivery.

Hyper text transfer protocol delivery In the HTTP delivered model the complete video file is downloaded by the client application and once the entire video has been downloaded the user is able to watch the video. The advantage of downloading the video before playback is that the file stored on the client's PC will be identical to that available on the server, thus the best quality video can be played to the user. With the entire video available at the client there is no issue with insufficient buffered data. The problem is that the user must wait for the complete video to be downloaded.

Hyper text transfer protocol progressive HTTP progressive is an alternative to downloading the entire file. In this model, the video is played while it is being downloaded. This removes the problem of requiring the downloading of the entire video before playback. However, HTTP progressive cannot stream every file format, which is unlike HTTP delivery.

HTTP progressive has the same problems as RTMP, namely the selection of the encoding rate and pauses in playback caused by packet loss.

2.2.2 User Datagram Protocol

UDP is the simpler one of the two commonly used transportation layer protocols [44]. When UDP packets are lost it is up to the application to handle. UDP does not provide any flow control or congestion control, and as such the application needs to implement suitable algorithms.

The advantage of UDP is that it delivers data to the application as soon as it arrives, since there is no buffering. With the lack of buffering and small amounts of packet processing, it means that UDP is a very light weight protocol compared to TCP.

2.2.2.1 Real-Time Transport Protocol

RTP is an application layer protocol in the TCP/IP stack which uses UDP as its transport protocol. The scope of RTP is to identify the payload type; provide packet ordering through sequence numbers; stream synchronisation through time stamping and monitor delivery and jitter [45]. These are achieved using the header shown in Figure 2.18, more details of which can be found in [1].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.18: H.264 RTP Packet Format [1]

Real-time transport protocol for H.264 In the context of H.264 video transportation, RTP can be considered as a video transport layer. In addition to the standard RTP header, for H.264 an additional header is prepended to the video data [46]. This additional header is termed the NAL Unit Type Octet, enabling the quick identification of the frame and partition type contained within the packet.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+-+-+-+-+-+-+-+-+
```

Figure 2.19: H.264 NAL header format [1]

NAL unit type octet The octet contains three parts, the Network Reference Indicator (NRI), the NAL unit type and a forbidden bit that must be one, shown in Figure 2.19 [28]. The NRI is a two bit field which indicates if the content of the packet is used as a reference by other slices. If the NRI field is equal to 00 then the packet can be dropped without causing errors in other frames [1, 28]. If the value of the NRI field is not equal to 00 then the packet should not be dropped. Thus when the content of the packet is a B slice the NRI is set to 00. For the remaining slices the NRI mapping typically follows those presented in Table 2.1.

Table 2.1: H.264 NAL unit type NRI mapping [1]

NAL Unit Type	Content of NAL unit	NRI (binary)
1	non-IDR coded slice	10
2	Coded slice data partition A	10
3	Coded slice data partition B	01
4	Coded slice data partition C	01

The NAL unit type is a five bit field identifying the contents of the packet, detailed in [28]. When data partitioning is used it is possible to use the NAL unit type to identify the partition type within the packet. The NAL unit type octet followed by the NAL unit is encapsulated within an RTP/UDP/IP packet for transportation over the network.

2.2.2.2 Unicast

The traditional method for packets to be delivered from the server to a destination is by unicast. In the unicast environment if multiple clients would like to receive the same video stream the same IP packet will need to be sent from the server to each of the clients. In the unicast case routers within the network only have to forward the packet. Figure 2.20 shows a sample network which transports only unicast traffic. The width of each network link is proportional to the bandwidth required such that each of the clients, C, within the network receive the packet stream sent from the server, S.

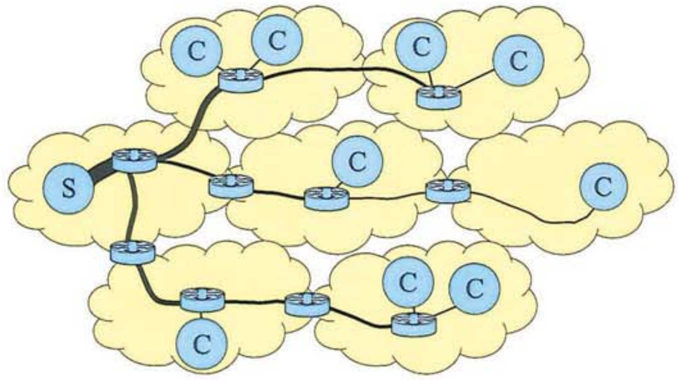


Figure 2.20: A sample unicast network

2.2.2.3 Multicast

Multicast provides an alternative to unicast. When multiple users request access to the same video stream, each IP packet only needs to be sent from the server using the multicast technique. Multicasting is only available to UDP because TCP is specifically designed to work with a single source and a single destination, although TCP for multicast has been investigated but currently its deployment is very limited [47].

If a client wants to receive a multicast stream it sends an Internet Group Management Protocol (IGMP) request to its local router. This local router then notifies the intermediate routers of the desire to receive the multicast stream. Once this is done the multicast traffic will be routed towards the new client.

Figure 2.21 is shown for comparison against Figure 2.20. Both figures have the same network topology, server and clients. Again, the width of each network link is proportional to the bandwidth require so that every client, C, receives the stream from the server, S.

Multicasting video on demand When videos are multicast, the client can only receive the packet sent from the server after it requests to join the multicast group. This is fine for receiving live broadcasts, but does not allow users to watch a program from the start unless they happen to connect at the exact right time.

A couple of solutions have been proposed to solve this problem. One method is to use multiple multicast streams, which are able to provide a near video on demand service. In this section we present Pyramid and patch broadcasting as examples of using multiple multicast

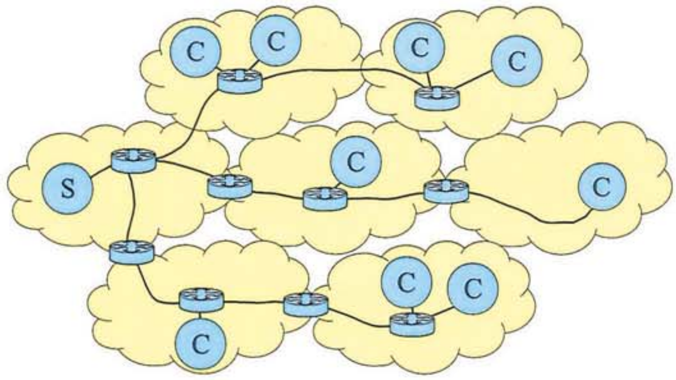


Figure 2.21: Sample multicast network

streams to provide near on demand videos. An analysis of multiple such techniques can be found in [48].

Pyramid broadcasting In Pyramid broadcasting a video is split into a number of segments. The duration of the segments is set in a way that the size of the current segment, Sd_n, is half of that of the next segment, i.e. $Sd_n = Sd_{n+1} \times 0.5$, where the last segment is the duration of half of the video. This segmentation can be done an infinite number of times, however this is impractical as this would require an infinite bandwidth. Instead, the number of segments is set to S, and the duration of the first segment is set to the same size as the second, i.e. $Sd_1 = Sd_2$. The choice of S is dependent on a number of factors. Since each segment must be streamed at twice the bandwidth of the video, the total bandwidth for the multicast streams is equal to $2 \times S \times B$, where B is the bandwidth required for playback. To reduce the total bandwidth requirement, the value, S, should be lowered. Conversely the maximum duration for which a client has to wait before it can start downloading is equal to ${}^{V}/_{2^{S-1}}$, where V is the total video duration. Therefore, the more segments there are the quicker the download will start [49]. An illustration of pyramid broadcasting is shown in Figure 2.22 with $S = 3$.

Patching A large number of schemes of multicasting a video constantly have been proposed [48]. Patching proposes to only stream videos on request. If additional clients choose to view a video then they can join the original multicast stream and this is supplemented with a patching stream so that the client can play the video back from the start, as opposed

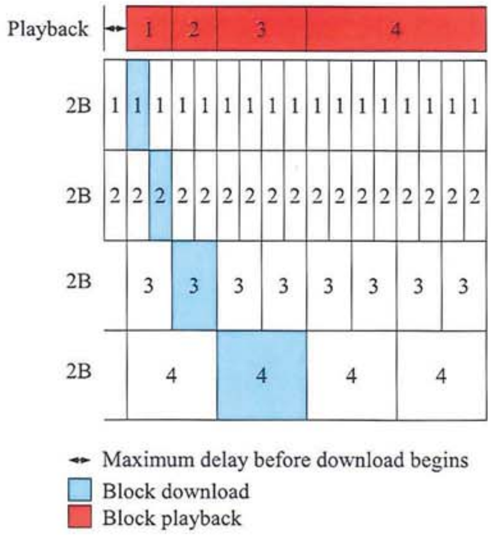


Figure 2.22: An example of pyramid broadcasting

to the current playback position of the original client [50]. The advantage is that there are no unwanted packets put onto the network, so reducing the load on routers. Additionally, this provides a more responsive video on demand experience as the requested video starts without a pause for the mutlicast cycle to restart.

Figure 2.23 shows a sample patching scenario where three clients request the same video but at different times. Client A initially requests the video and a multicast stream starts, termed stream A. When client B requests the video it also starts downloading stream A and receives a patching stream, called stream B, that provides the video between the start and the point at which it joints stream A. The same happens when client C joins. It receives the stream B and its own patching stream, termed stream C. Once client C has got to the point at which it joins the patching stream B, its own patching stream becomes defunct. Client C then downloades stream A in addition to Stream B. Figure 2.24 shows the periods within the video played to the client and downloaded from the different streams.

Multicast issues There are a number of reasons why multicast video on demand has not taken off. Firstly, all on demand multicast techniques require more bandwidth available to the client than is required for video playback. Secondly, unlike unicast where an IP address identifies an individual computer, multiple computers can operate using the same multicast IP address.

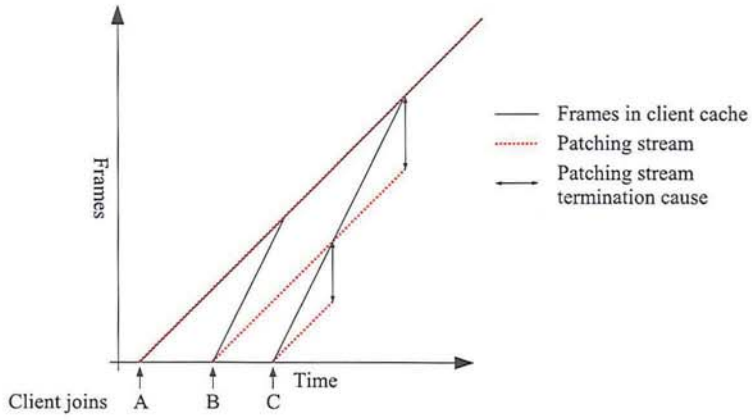


Figure 2.23: Example of the patching multicast technique

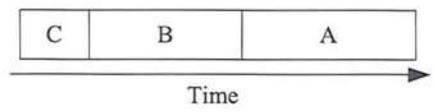


Figure 2.24: For client C, the period of video download from each stream

However, the most significant problem is that IP multicasting is quite often unavailable, either because it is disabled by administrators due to security issues or because of limited deployment of multicast enabled routers due to the added complexity to each router. Even in networks where multicasting is available it is regularly filtered out once it is passed to a neighbouring network. To overcome this problem research efforts have been made using unicast communication to link together multicast enabled networks. However, this scheme also has security issues and is no longer deployed [51].

2.3 Quality Measurement and Error Concealment

There are two categories of quality measurement for received videos, objective and subjective. Using the subjective measurement method, a large number of users are asked to rate the quality of the video received. This measurement is very labour intensive and so objective quality measurement techniques are more common, since they can be computed.

In this section, we present two objective quality measurements: peak signal to noise ratio (PSNR) and negative sobel difference (NEGSOB). These measurements are used to measure the quality of the decoded video compared to the original video. In a multimedia environment, where both video and audio data co-exist, additional tools can be used, such as Perceptual Evaluation of Video Quality (PEVQ) which gives an indication of the overall user experience. In our work we do not consider a multimedia environment and restrict our analysis to the quality of video.

For the quality of frames to be measured, complete frames are required. When losses within a video stream occur error concealment must be executed to complete the missing portions of the frame. Error concealment is an important technique to mention since the quality of the error concealment schemes will affect the measurement results and the quality perceived by the end user. The discussion of this technique will follow the introduction of the two previously mentioned quality measurement schemes.

2.3.1 Peak Signal to Noise Ratio

To measure the quality of an encoded frame, the Peak Signal to Noise Ratio (PSNR) is the de facto standard. Traditionally, PSNR is used to compare the results of compression with the original frame, as shown in Figure 2.25 as compression quality. In a lossy environment, PSNR is also used to measure how similar the received picture is to the original. In this environment, the quality of compression, transport and concealment resulting in the received frame are compared against the original.

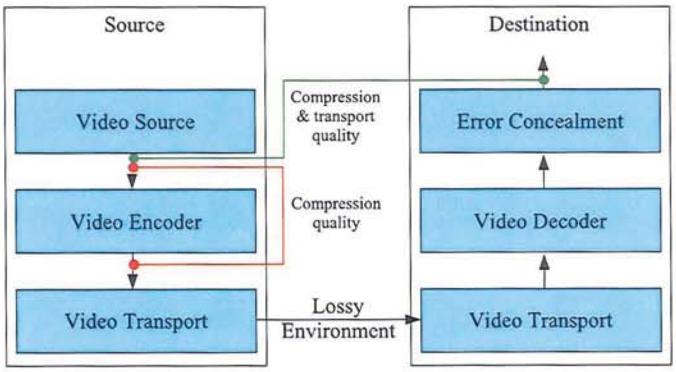


Figure 2.25: Image comparison options for the calculation of PSNR

PSNR is calculated using the formula

$$PSNR = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{2.3}$$

where MAX_I is the maximum number of pixel values for the decoded frame and MSE is the mean square error (MSE) of the compressed image, which is defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j)-K(i,j))^2 \tag{2.4}$$

where $I(i,j)$ is the luminance value for the pixel located at coordinate (i,j) within the original frame. Where $0 \leq i < m$ with m equal to the width of the frame in pixels, similarly $0 \leq j < n$ with n equal to the height of the frame in pixels. $K(i,j)$ is the luminance value for the pixel located at coordinate (i,j) within the frame to which the original is to be compared. The widths and heights of K and I must be the same. The value of $K(i,j)$ is equal to $I(i,j)+\varepsilon(i,j)$ where ε is the error introduced through the compression and transportation of the video frame which is present after decoding the frame.

PSNR is the comparison method mostly used by researchers. However there are a couple of alternative methods that can be split into two groups, the direct pixel-to-pixel comparison and the calculation of an intermediate set of values before comparing the frames. Of the pixel-to-pixel comparison types PSNR is the most popular as it is more established than newer comparison methods, such as Structural Similarity (SSIM). The non-direct pixel-to-pixel comparisons are less common, but for completeness we include the negative sobel difference (NEGSOB) as an example of the non-direct pixel-to-pixel approach.

2.3.2 Negative Sobel Difference

NEGSOB uses sobel edge detection to aid in the identification of artefacts. Using a block based compression tool, as is the case with H.26x, the edged of the blocks can be present in the decoded frames. NEGSOB tries to quantify these edges which should not be present in the frame.

The sobel value for each pixel of a frame is calculated using two matrices [52, 53], one for detecting edges in the horizontal direction and the other in the vertical direction. Using these matrices, we are able to calculate the values for $|G_y|$, the detection result of edges in the vertical direction, and $|G_x|$, the detection result of edges in the horizontal direction. For example, given an original image $O(i,j)$ where $0 \leq i < m$ with m equal to the width of the frame in pixels, similarly $0 \leq j < n$ with n equal to the height of the frame in pixels. A 3 by 3 region, $A(i,j)$, is selected such that

$$A(i,j) = \begin{bmatrix} O(i-1,j+1) & O(i,j+1) & O(i+1,j+1) \\ O(i-1,j) & O(i,j) & O(i+1,j) \\ O(i-1,j-1) & O(i,j-1) & O(i+1,j-1) \end{bmatrix} \tag{2.5}$$

To prevent $A(i,j)$ including pixels outside of the image, the values of i and j must be kept within the range $0 < i < m-1$ and $0 < j < n-1$. Figure 2.26 shows a graphical representation of the sobel calculation mapping from the original array of values to the resulting array.

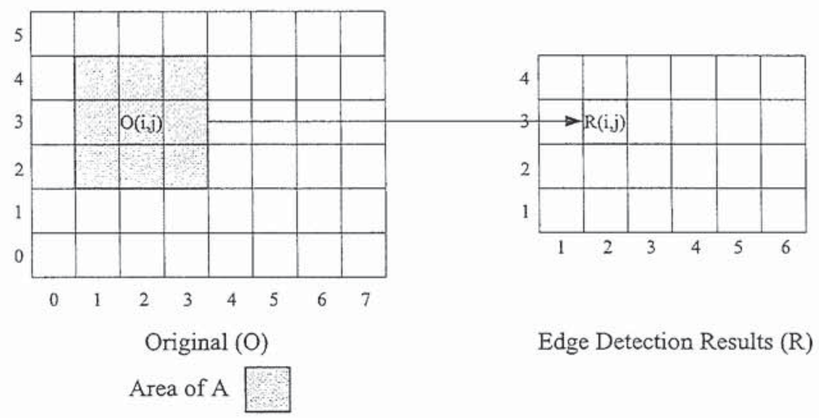


Figure 2.26: NEGSOB calculation mapping

The calculation of the corresponding $R(i,j)$ value is then computed as follows

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \times A \tag{2.6}$$

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \times A \tag{2.7}$$

$$R(i,j) = |\sqrt{G_x^2 + G_y^2}| \tag{2.8}$$

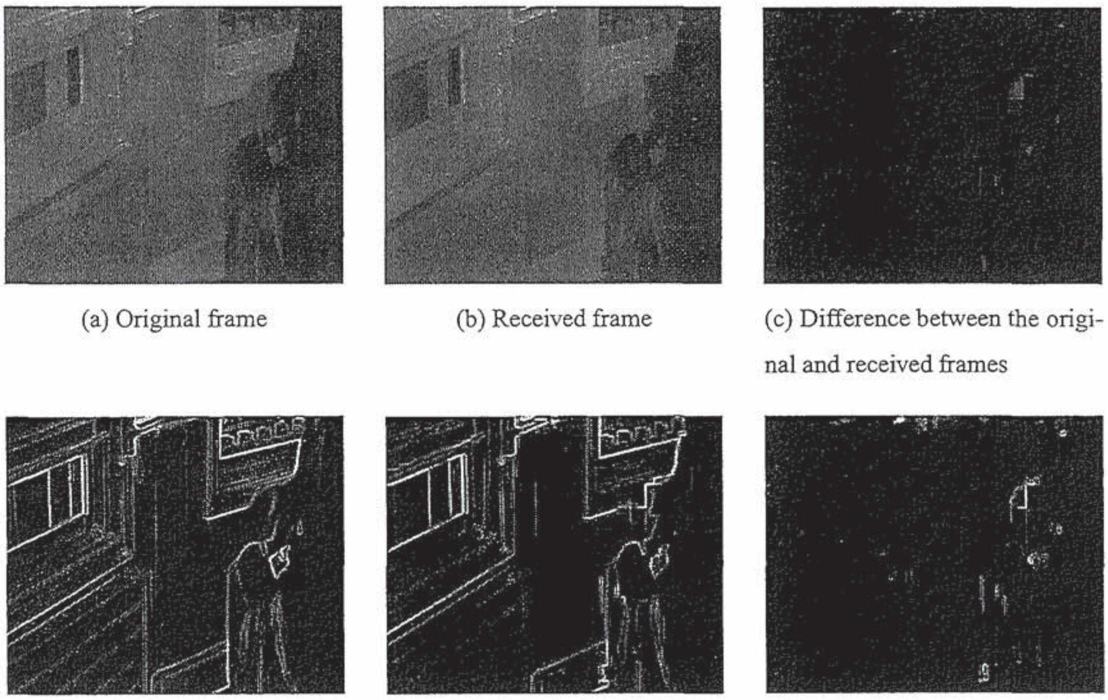
The NEGSOB value for the image is given by

$$\frac{1}{(m-2)(n-2)}\sum_{i=1}^{m-2}\sum_{j=1}^{n-2}[R_O(i,j)-R_C(i,j)]_{np} \tag{2.9}$$

where R_O is the edge detection result for the original image, R_C is the edge detection result for the comparison image, $0 \leq i < m$ with m equal to the width of the frame in pixels, similarly $0 \leq j < n$ with n equal to the height of the frame in pixels and $[x]_{np}$ is defined as

$$[x]_{np} = \begin{cases} 0 & \text{if } x \geq 0 \\ x & \text{if } x < 0 \end{cases} \tag{2.10}$$

As an example, Figure 2.27 shows the sobel edge detection of two images and their difference. Figure 2.27a shows an original frame sent from a server to a client. After packet loss and error concealment the image shown in Figure 2.27b is displayed to the client. Figure 2.27c shows the difference between the original and received frames. Figures 2.27d and 2.27e show the results of sobel edge detection on the original and received frames, respectively. Figure 2.27f shows the difference between the sobel values on a pixel by pixel basis.



(a) Original frame

(b) Received frame

(c) Difference between the original and received frames

(d) Negative sobel filtered original frame

(e) Negative sobel filtered received frame

(f) Negative sobel difference

Figure 2.27: A demonstration of the results of sobel differences in a lossy environment

2.3.3 Error Concealment

There are many different error concealment algorithms [54–56]. These algorithms are able to conceal missing parts of the frame with varying levels of success. It is generally easier to conceal areas with little detail but this is not the same for complex areas.

The smallest element of a video is a macroblock. When we stream videos over an IP network, it is likely that multiple macroblocks or potentially even large portions of the fame will be lost at the same time.

Different concealment schemes, have varying levels of complexity. Choosing a proper error concealment scheme needs to consider the trade off between complexity and the noticeability of errors. This is because generally more complex schemes produce frames in which the macroblock losses are less noticeable.

2.3.3.1 Spatial Concealment

The computationally simplest way to conceal losses is to fill the erroneous macroblocks with a single colour. When using this simple scheme it is trivial to notice the parts of the screen which have been lost or corrupted. In an environment where the player receives nearly no loss then spatial concealment might be acceptable. In lossy environments, however, more intelligence is required.

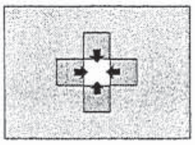


Figure 2.28: Spatial error concealment

A more intelligent decoding scheme would be to use the areas neighbouring the damaged macroblocks to predict the missing content. This is termed spatial concealment, illustrated in Figure 2.28. Spatial concealment works well in areas of low detail and has the advantage of being able to decode the video on a frame by frame basis.

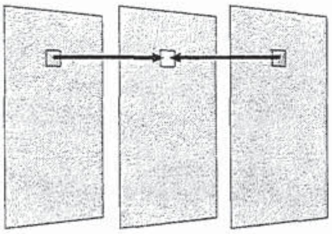


Figure 2.29: Temporal error concealment

2.3.3.2 Temporal Concealment

Within a video stream, there is likely similarity between the current frame and those both before and after it. Temporal concealment, shown in Figure 2.29, takes content in both the previous and later frames and extrapolates between the two to fill the missing area. With slowly moving frames, temporal concealment works well, even with blocks containing high levels of detail. If the losses occur in a region where large amounts of motion are present then the concealment scheme performs poorly.

It is possible to merge both the temporal and spatial concealment techniques using a three-dimensional extrapolation process. This results in an improvement in the quality of the concealed frame, but at the cost of increased processing requirements [55].

2.3.3.3 Frame Copy Concealment

The temporal and spatial schemes discussed are designed to conceal losses within a frame. In the environments where complete frames are lost, these schemes can not be directly applied. Frame losses are possible when streaming video over a packet network, such as the Internet, as a slice might all be sent in the same packet. Two frame concealment techniques, frame copy and Motion vector copy [54], are introduced to deal with this problem.

With frame copy concealment the previous reference frame is used to replace the lost frame. To the viewer this might just appear as a reduction in the frame rate but has the benefit that no artefacts are displayed for the lost frame. However, since the lost frame is likely to be used as a source of reference for other frames, this will cause incorrect data to propagate. This propagation of errors will create artefacts in future frames.

2.3.3.4 Motion Vector Copy Concealment

Motion vector copy aims to reduce the effect of errors propagating to future frames by approximating the motion vectors which would have occurred between a lost frame and the predicted frame preceding the lost frame. This is achieved by taking the motion vectors from the previously decoded reference frame and scaling them depending on the distance of the lost frame from the reference frame.

Motion vector copy has been shown to outperform frame copy as a concealment technique for lost frames at a range of different encoded bit rates [54, 56]. Additionally, motion vector copy also outperforms at nearly all packet loss rates. The authors of [54] show that using motion vector copy can perform up to 6dB better than frame copy when the stream experiences a packet loss rate of 20%.

2.4 Conclusion

H.264 is currently the most efficient video codec, providing large performance gains over preceding video encoding standards. These improvements include halving the data rate for coding a video with MPEG-4 part 2 with the same fidelity compared with MPEG2.

H.264 data partitioning provides the ability to separate data of different importances into different packets with little loss in compression efficiency. This is in comparison to SVC where there is an increase in the bandwidth requirements of 10%.

Two transport protocols have been presented, TCP and UDP. TCP treats all packets, termed TCP segments, with equal importance. When working with streams of unequal importance, such as H.264 data partitioned video streams, TCP would still treat all packets equally and automatically request retransmissions for any lost data and then reduce its data rate. In the remainder of this thesis, we use UDP since this allows greater control over the individual packets. We set out to use UDP in a unicast environment due to the limited support for multicast within the Internet.

We have presented two different measures for video quality. Even though PSNR is designed for compression loss, it is regularly applied to any difference between two images. Hence, we use PSNR to measure frame quality, in the same way adopted by the research community.

Finally, we have discussed various error concealment schemes, of the concealment schemes presented motion vector copy concealment produces the best results. However, in this thesis an exhaustive investigation into concealment techniques has not been conducted.

Chapter 3

Quality of Service for Video Streaming

The Internet Protocol is a best effort packet switching network. As in any best effort networks there is no quality of service (QoS) guarantee that a packet will be routed successfully from the source to the destination. In the Internet environment, QoS problems are caused by limited bandwidth and buffering capacity along with equipment failure, which result in packet loss, delay and delay variation (jitter) [57].

Different applications require different levels of QoS. For example, tasks such as checking email or system updates, which are background processes and do not have direct user interaction, would require a much lower QoS than video or audio streaming. With video or audio the users experience will be dependent on the QoS of the network.

The issue of delay is only related to the duration between the user selecting a video and playback starting, after which the time taken for a packet to cross the network has no effect on the quality of service received by the user. Once the streaming of the video has started the issues of packet loss and jitter are far more important than the latency.

When jitter is present, packets can arrive after the point which they are required for playback, and will have the same effect as a lost packet. To reduce the effect of jitter the video client software can buffer some of the video stream. The problem with buffering is that the larger the buffer is the longer it will take to fill, resulting in a longer duration between the user selecting a video and playback starting. If the buffer is bigger there will be fewer occasions when the jitter exceeds the buffered duration.

The third QoS parameter is packet loss. As discussed previously in Section 2.2, there are two ways to transport video over an IP network using either the reliable TCP or unreliable

UDP. When TCP is used for video streaming the effect of packet loss can be a pause in playback of the video at the client side. However, when UDP is used there will be no pauses in playback due to the loss of packets. Instead, there will be a reduction in the quality of the video displayed to the client for a period of time, which is a result of the removal of temporal redundancy when the video was encoded. When a frame is decoded with errors, due to a lost packet, these errors will exist in the reference picture used to decode future frames. As such errors could be displayed to the user until the end of the GOP.

It is a common misconception that within the TCP/IP stack, shown in Figure 3.1, that if congestion occurs within the network the network layer will route around it [58]. One of the major purposes of the network layer is to provide independence from the intricate details of routing or switching of packets across the network [59,60]. The transport layer is responsible for ensuring that the packets arrive. Therefore, managing the effect of congestion or loss is a transport layer issue.

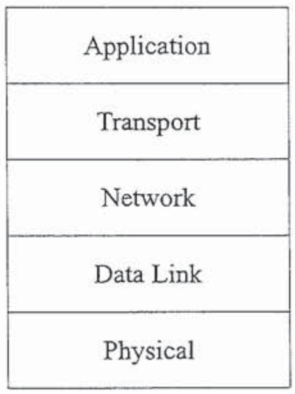


Figure 3.1: TCP/IP Stack

It has been seen that when transporting video through a lossy network even a relatively small percentage of packet losses can cause a large percentage of frames to be displayed wrongly. For example a loss rate of just 3% of the packets transmitted could result in the erroneous display of 30% of frames in an MPEG2 stream [61].

When comparing wired transmission, delivered by fibre optics or copper, against wireless transmission it is commonly agreed that for the same distance the wired network will be practically loss free compared to the wireless network. In a wired link the greatest effect on QoS is not the wired network itself but the routing between different physical network links.

In a wireless link QoS is affected greatly by the quality of transmission, thus special link layer techniques for enhancing QoS can be utilised.

3.1 QoS in an IP Backbone Network

Over the years QoS schemes within the Internet have changed and developed. The initial QoS framework included in the Internet Protocol version 4 (IPv4) was not widely implemented in routers. Recently new QoS schemes have superseded the original provided by IPv4, and currently there are two frameworks, Integrated services and Differentiated services, which are supported by some routers. An overview of both of these techniques follows.

3.1.1 Integrated Services

Integrated services (IntServ) provide a fine-grained end-to-end QoS system enabling flow specific QoS arrangements. This is achieved through the use of the Resource Reservation Protocol (RSVP) [62,63] which allows an application to reserve resources within the network for a packet stream. A packet stream is defined as a sequence of packets, which have the same source, destination and QoS requirements. It should be noted that a packet stream is simplex. Although RSVP operates at the transport layer, it does not transport any data itself.

IntServ creates two new service classes in addition to the existing best effort service, which are termed guaranteed service and controlled load service. The controlled load service class makes networks appear as though they are underutilised, i.e. having a low loss rate and small delay [64]. This service class provides no explicit guarantees on loss or delay. The alternative to controlled load is guaranteed service which provides a limit on the maximum delay a packet can experience while it crosses the network, thus reducing the maximum amount of jitter and eliminates buffer overflow guaranteeing the arrival of a packet at its destination [65].

To enable RSVP a client makes a request to the local RSVP process. This RSVP process then forwards the request to all nodes along the reverse path back towards the source. Once an RSVP request is received at a router two checks are performed; that the router is able to meet the requested demand and that the user has suitable permissions to make such a request. If successful, the streams details are stored in the packet classifier and in the link layer interface to permit the requested QoS service.

IntServ is perfectly suited for the QoS requirements of a small network. However, due to its requirement that stream specific data must be stored at each router and combined with the processing required for each packet, it scales poorly. Due to its poor scalability IntServ is rarely implemented on the Internet.

3.1.2 Differentiated services

Whereas IntServ operates on an end-to-end stream basis, Differentiated Services (DiffServ) operates on a per traffic class per hop basis, where each packet is assigned to a packet class and this class is stored in the packets header. By operating on a packet basis, routers are not required to store stream specific settings and as a result this scheme scales up to Internet size networks, which IntServ is unable to achieve. Additionally, DiffServ has the benefit that it can be selectively deployed at bottlenecked locations and does not need to be deployed at every router. The assignment of packets to classes can either be policy-based and done at a router on the edge of the network or by the packet source. For security reasons it is wise to reset the packet assignment at the edge router to prevent applications from acting unfairly.

The disadvantage of DiffServ is that there are still no QoS guarantees. With IntServ, resources are reserved for a particular stream and streams are declined access if there is insufficient available resources, whereas with DiffServ there is no set-up phase. Without this set-up phase the routers are unable to reserve dedicated resources for the selected stream. As a consequence intermediate routers might not have sufficient available resources at the instant the packet arrives to handle the request and the packet might be dropped.

The IP layer of the TCP/IP stack, shown in Figure 3.1, enables end-to-end routing of packets across the Internet. The IP packet header is shown in Figure 3.2, of which we are interested in the Type of Service (ToS) field.

The ToS field was originally designed to allow an application to select options for the packets routing across the network, including low delay or high throughput. The ToS field was not widely implemented and so it was later decided to redefine the ToS bits as the Differentiated Services (DS) field. The DS field is further sub divided into the Differentiated Services Codepoint (DSCP) and the Explicit Congestion Notification (ECN) field [66], as shown in Figure 3.3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.2: IP Packet Format [4]

```
    0     1     2     3     4     5     6     7
 +-----+-----+-----+-----+-----+-----+-----+-----+
 |          DS FIELD, DSCP           | ECN FIELD |
 +-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 3.3: Differentiated Services Format [5]

The DSCP field can contain one of 64 different values indicating the importance of the packet [67]. When a packet arrives at a router it will be assigned to an Access Category (AC), depending on the DSCP field within the packet.

Although the DSCP values are standardised [67] (but not all equipment complies with the standard [68]), there is no standard definition of how to handle these packets at routers. Techniques such as priority queuing, rate queuing, active queue management and traffic conditioning could be used but their selection is operator dependant. Additionally, which DSCP values get assigned to which class and even how many classes should operate at the router are decided by the operator.

3.1.3 Path Diversity Utilisation

Although IntServ is able to overcome the issues of congestion and router update neither are achieved by DiffServ. The issues of congestion and router updates come about because the network layer is only concerned with the issues related to routing and do not monitor the condition of each link. The guarantees of packet delivery are transport layer issues.

Derived from the fact that the Internet was originally developed with funding from the United States Defence Advanced Research Projects Agency (DARPA), the Internet was designed to be able to cope when parts of the network were destroyed or disabled [69]. It is this original design criteria that between any two Points of Presence (PoP) on the Internet there is likely to be multiple different paths available.

Although the Internet has developed differently from its original aims, recent research [70] has shown that the original design principle of redundancy still influences the Internet today. The authors of [70] looked at Sprint's (a large US Internet Service Provider (ISP)) internal network and found that for a large percentage of PoP pairs there are multiple partially diverse paths between them. The authors also looked at the diversity across multiple networks and highlighted that diversity is not fully exploited. If applications were able to utilise the multiple available paths then this would give options for load balancing and reducing the packet loss rate, to improve link recovery rate and provide fault tolerance [71].

To enable inter-domain routing (also termed inter autonomous system (AS) routing) within the Internet, each AS needs to be aware of which other networks connect to which. The procedure of learning how ASs interconnect is through the use of the the Border Gateway Protocol(BGP) [72]. BGP operates by sending messages to other BGP nodes, which include the prefixes that the router has routes towards [73]. The design of BGP is to converge onto a route [74]. Convergence is important to prevent packets from being routed in a circular manner and never reaching their destination. Since BGP converges onto a route, a lot of the underlying redundancy is abstracted. There are a number of ways around this abstraction; source routing, router alteration, multihoming or path diversity.

Another issue with BGP is that when routing tables are updated this causes temporary routing issues. In the investigation of Voice over IP (VoIP), researchers have found that the effect of BGP updates results in unintelligible voice levels for over four minutes [75]. A number of researchers have looked into using the path diversity technique with VoIP [76–78], but far fewer have looked into the effect of path diversity on video streaming.

BGP is notoriously slow at recovering from link failure. Route convergence takes an average of three minutes (can take up to fifteen minutes) [79, 80], during which packets can be lost or delayed.

It is common for AS routes not to be advertised, because of peering and transit relationships between ASs. Peer relationships are usually put in place so that clients of one AS have quicker access to the resources of the second AS, but the connection between the two is not advertised to any other ASs, meaning that no traffic will enter the first AS trying to get to the second. With a transit agreement two ASs are connected, as with the peering arrangement, and advertise the availability of the route to other networks [6].

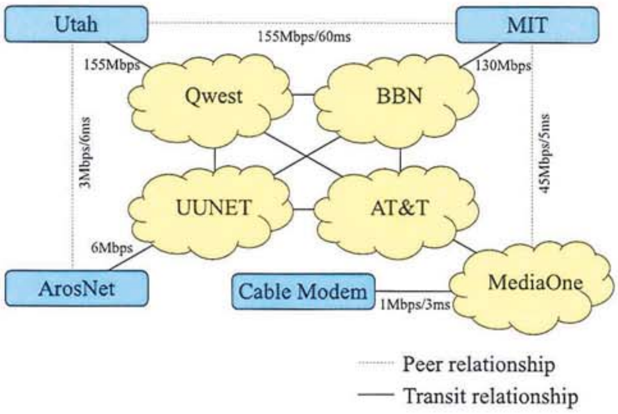


Figure 3.4: Example of peering networks [6]

An example topology is shown in Figure 3.4. If MIT wants to contact ArosNet it has to go via BBN and UUNET, although there is also a route via Utah that can never be used as only Utah's routers are made aware of it and not MIT's.

3.1.3.1 Source Routing

One of the earliest methods of utilising path diversity was with the option for loose and strict source routing. These options were included within IPv4 [4].

With strict source routing the packet header contains a completely predefined sequence of routers for the packet to pass through. When the packet arrives at each router the next IP address from within the header is used. The benefit of strict source routing is that an exact route through the network can be achieved. However, the source must have a detailed knowledge of the underlying network topology. Maintaining a map of the Internet would be an impossible task. However, loose source routing can provide some of the benefits of strict source routing but without the requirement of maintaining a complete map of the underlying topology.

The difference between strict source routing and loose source routing is that in the latter only some of the intermediate routers are specified, as opposed to all of them in the former.

Between the specified routers the packet is routed as any normal packet would be. This means that if a nonspecified intermediate router is no longer available then existing routing techniques will re-route the packet, and will not require the source to update its topology map. If the specifically selected routers become disconnected then the application will still need to select a replacement. Loose source routing requires less management by the source application, but this is at the cost of reduced control over the packets routing.

Figure 3.5 shows an example of source routing, which is included for comparison against other diversity schemes presented later in this chapter.

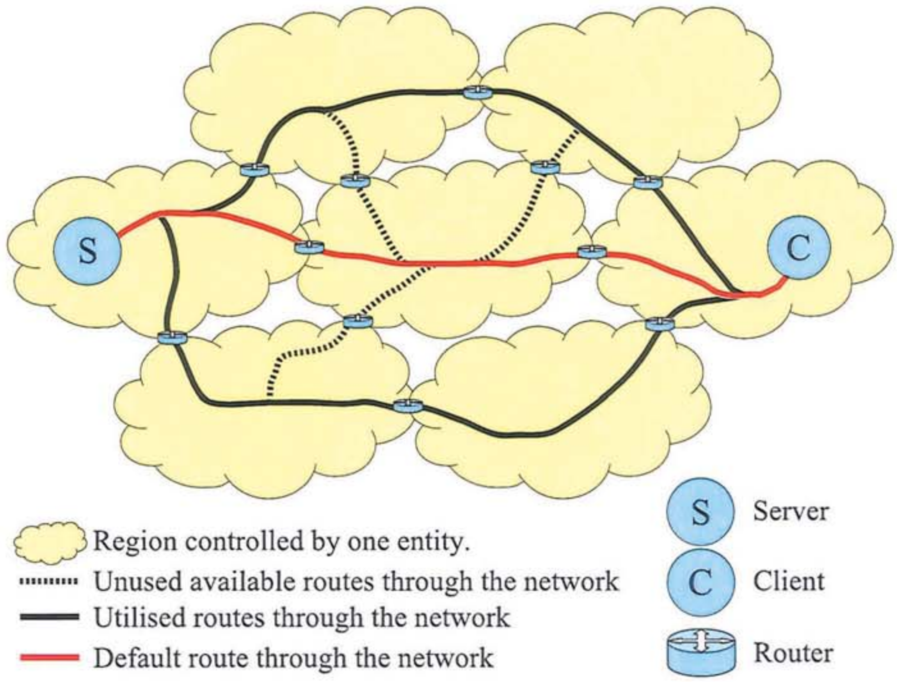


Figure 3.5: Illustration of path diversity using source routing

For both strict and loose source routing to perform optimally, the underlying network topology would need to be known by the server. Due to the growth of the Internet and the fact that each autonomous system is self managing, no complete network topology exists. A number of techniques have been proposed to compile partial network topologies [81, 82]. The techniques proposed in these papers could provide sufficient topological information for source routing to be beneficial, however the use of source routing is limited since many routers block packets containing the source routing option because of security concerns [83].

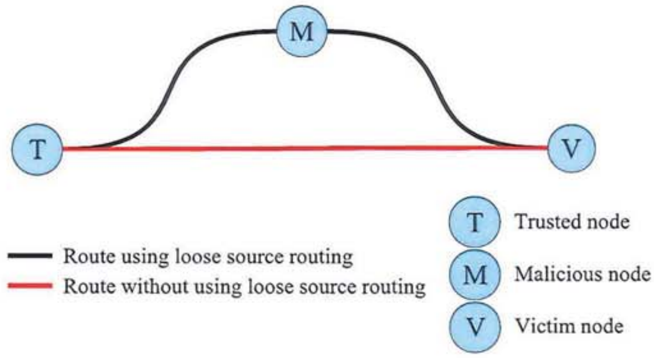


Figure 3.6: Illustration of source routing security issue from an intermediate node

Figure 3.6 gives one example of a source routing security issue, where T is a trusted node, V is the victim node and M is the malicious node [84, 85]. If node M creates a packet, within this packet it sets the destination as node V and can also set the source as node T. When the packet arrives at node V, node V thinks that this packet has come from node T, and then replies to node T. Without source routing the packet would unlikely travel back to node T via node M. However, when node M created the packet it set the source routing path such that it appeared as though the packet left node T and went via node M to node V, it would thus make sense for node V to reply via node M. When the packet then goes back via node M, node M is able to capture the reply and discard the packet before it gets back to T. This method gives node M access to any services available to node T, which might include services restricted based on its IP address.

Loose or strict source routing would provide one of the best ways to achieve path diversity. However, because they have been widely disabled a number of other proposals have been made. These proposals include multihoming, overlay networks and changes to network routers.

3.1.3.2 Router Alteration

A number of proposals have been made to adapt BGP [73] for better utilising diversity. The authors of [86] propose a method where an application can negotiate with an extended version of BGP to select paths which avoid certain ASs, thus enabling an application to create multiple streams crossing the network to take different paths. One of the problems with such a scheme is that it requires an additional stream set-up delay to negotiate the routing of packets.

The authors of [87] propose a scheme where different TCP streams are routed across different available paths. The paper shows that an increase in throughput can be achieved, which might not be true in reality as they model links with different bandwidths. The paper does not analyse the scenario where the links are of the same bandwidth.

In addition the authors of [88] present a scheme where access routers are positioned at the source and destination AS with transit routers positioned in between. The access routers maintain multiple paths from the source to the destinations. In this way they can select the best quality path at any given point in time and switch as network conditions change. The problem with this scheme is that it would require a large amount of storage and processing to maintain the round trip times (RTT) to each destination AS.

Another scheme that has the potential to introduce redundancy through router alteration is that presented in [89]. This scheme provides the benefits of source routing but with a more secure key mechanism. Within a network, packets containing an authorised key, properly explained in [89], are allowed to be routed via internal nodes. This provides the benefit that only trusted source networks or nodes would be allowed to route packets into certain networks.

The Bananas framework [90] provides an alternative to source routing, where instead of specifying intermediate routers by their IP address you provide an outgoing interface number. The advantage of this scheme is that the amount of space required to include a port number in a packet is much smaller than the space required to include a complete IP address. The framework includes a method to validate a route, without which it is possible for a packet to be caught in a loop and never reach its destination.

There are a number of reasons for the limited interest in alterations to routers. Firstly, anything which makes routers more complex will increase their cost and potential for firmware bugs due to increased complexity and thus failure. Secondly, router alteration is trying to make routers work at the transport layer and not just the network layer. Through the introduction of such changes, the network layer also has the potential to break existing protocols, such as TCP. TCP requires an accurate measure of the RTT to be taken [91] and if the underlying network keeps changing then an accurate RTT value can't be established.

3.1.3.3 Multihoming

Another way to introduce diversity into the network is by connecting to multiple diverse ISPs. Packets can then be put onto the different networks in the hope that at least part of the paths taken will be independent of the other paths. An illustration of multihoming is shown in Figure 3.7.

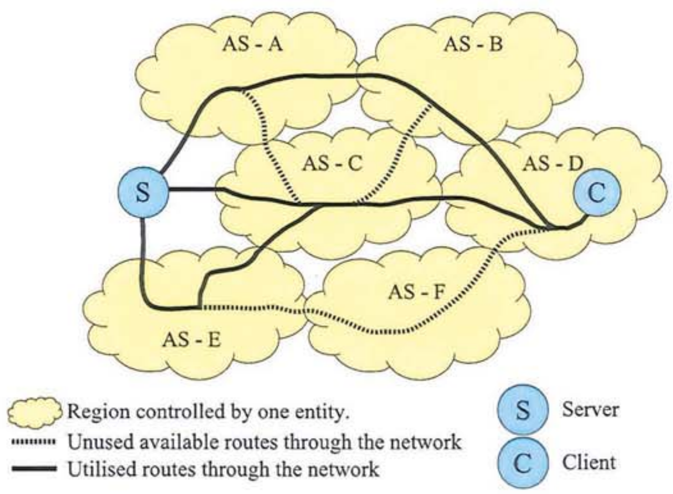


Figure 3.7: Illustration of path diversity using multihoming through connections to three ASs

In Figure 3.7 we see a server, S, connected to three networks, these are AS-A, AS-C and AS-E. Each AS has its own routing policy when routing a packet from S to its destination, C. Packets sent to AS-A first are routed through AS-B and AS-D to get to C. Packets which are sent to AS-C are routed via AS-D. Finally, Packets sent to AS-E first are routed to AS-C which then routes packets to AS-D. The server only controls the first AS which the packet will be routed onto, as such for multihoming to produce benefits careful ISP selection must take place [92] based on their peering agreements.

If connections were made to ISPs which have peering agreements then it is more likely that the packets put onto the different networks will be routed together earlier in the path than if the ISPs do not have a peering arrangement. To ensure that the diversity is not reduced over time the monitor of inter peering arrangements should be considered.

Investigations into mutlihoming through emulation have been conducted by the authors of [93]. They show that improvements in the response time can be achieved in 65% of cases when three ISPs are used with passive network measurements.

The authors of [92] conduct a number of experiments to investigate the benefits of multihoming using the Akamai network, finding that with connections to two ISPs that a 25% improvement in performance, as measured by the turnaround time, is achieved in three out of their four tests. The turnaround time is the duration between the end of the request, in the case of a HTTP request, and the start of the response from the server. Additionally, the authors present results showing that connecting to more than four ISPs produces little additional benefit.

3.1.3.4 Overlay Networks

An overlay network creates partially diverse paths by recreating loose source routing at the application layer of the TCP/IP stack, achieved through the introduction of relay nodes within the network. Figure 3.8 shows an example of an overlay network containing two relay nodes, creating three paths through the network. Figure 3.9 shows the TCP/IP stack of an overlay network.

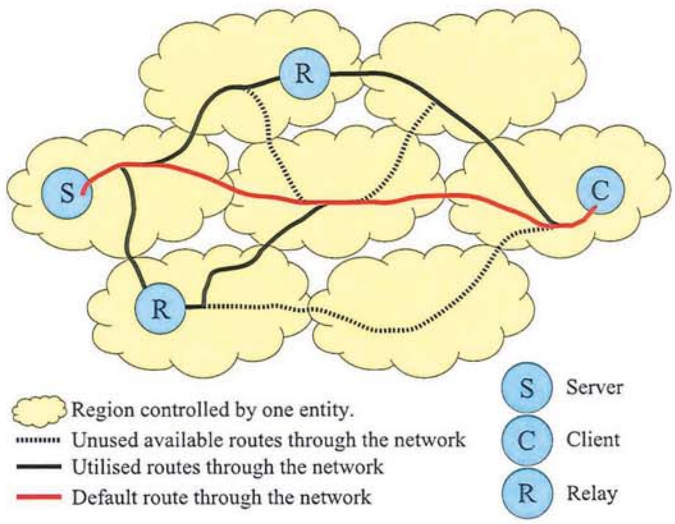


Figure 3.8: Illustration of path diversity using an overlay network

A number of papers on the topic of path diversity have examined video distribution [94–96], but these have not investigated the effect of increasing the number of paths beyond two.

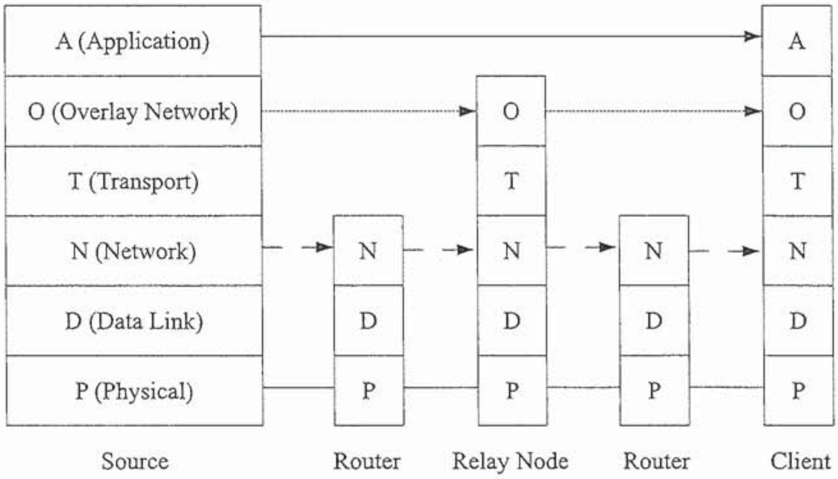


Figure 3.9: TCP/IP Stack with overlay network layer

Other work on this issue is focused on reducing packet losses with algorithms for allocating packets to each of the two paths, aiming to reduce the total number of losses [97].

The idea of using overlay networks is not new and it has received some attention. In [71] the authors provide an overview of both path diversity and content delivery networks and look at overlay networks in the wireless environment. The authors of [6] create an overlay network and use active probing to monitor network characteristics. The use of active probing allows the application to select the most appropriate path. The problem with active probing is that it has a significant overhead and could unfortunately aid in the creation of congestion.

In addition to the investigations into the methods of utilising diversity, some authors have looked at methods for maximising throughput, achieved by altering the amount of data sent down different paths [95, 98]. The methods proposed in these papers are better than the active probing techniques used in [6], in terms of bandwidth efficiency, but they still require additional feedback to the server. There are some environments where a feedback channel is not desired as this increases the amount of processing required and can limit the number of acceptable clients [99].

3.2 QoS in Wireless Local Area Network

Figure 3.10 shows the TCP/IP stack and a breakdown of the data link layer. The stack is designed so that higher layers, such as the application layer, do not need to know the intricate detail of the lower layers, such as the physical layer. Therefore it is possible to replace the data link layer and physical layer with a wireless protocol. There are a number of wireless standards which could be used to replace the Data Link and Physical layers, including the Institute of Electrical and Electronics Engineers (IEEE) standards 802.11 [7] and 802.16 [100], along with High-Speed Downlink Packet Access (HSDPA) [101] standardised by the 3rd Generation Partnership Project (3GPP).

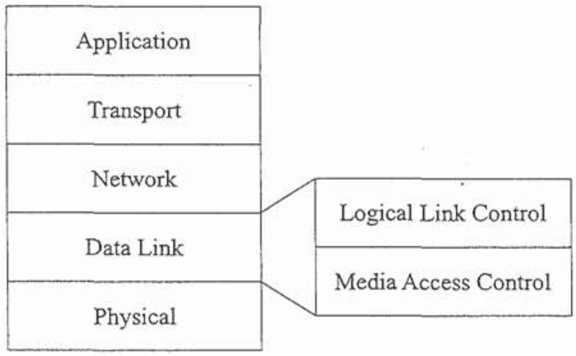


Figure 3.10: TCP/IP Stack with Data Link Layer breakdown

IEEE 802.11, IEEE 802.16 and HSDPA target different markets, hence could all co-exist in the future. IEEE 802.11 is designed as a wireless equivalent to their popular wired Ethernet standard (IEEE 802.3 [102]) for Local Area Networks (LAN). LANs are generally small networks covering a single room, floor or building, which generally connect to a wider network such as a Metropolitan Area Network (MAN) for connectivity to additional resources.

IEEE 802.16, also known as Broadband Wireless Access, is primarily a Wireless Metropolitan Area Network (WMAN). MANs are designed to connect a number of LANs to a Wide Area Network (WAN). These LANs are themselves likely to be wireless, in which case they will most likely be 802.11 networks.

IEEE 802.16e [103] extends IEEE 802.16 to include the option for mobile broadband wireless access and competes directly with the currently deployed HSDPA systems. It is dependent on the future pricing of such services, which will dictate if consumers are happy for each laptop to have its own mobile data subscription. Otherwise, wireless broadband services, provided by either 802.16e or HSDPA, are likely to be shared between laptops with the use of an 802.11 network. There are already a number of HSDPA/802.11 access points in commercial production, but it depends on the wide spread take-up of the technology to dictate if the last link to the client is 802.11, 802.16e or HSDPA.

In this thesis we look at 802.11 networks, because firstly it is possible to use 802.11 networks as the last hop either to a wired network or onto a wider area wireless network, and secondly we have recently seen a large increase in the adoption of 802.11 within homes and businesses [104]. The WLAN technology has been widely implemented in laptop computers and is increasingly being used in other devices, such as phones and cameras.

The problem with this technology is that for the same distance the loss rate of a wireless network compared to a wired network is much higher. There are a number of causes of this. The 802.11 protocol series operate in licence exempt frequency bands; 802.11b and g operate at 2.4GHz and 802.11a and n operate at 5GHz. In these licence exempt bands interference can be experienced from other devices operating in the same band, which could be other WLAN devices, microwave ovens or wireless phones. Due to the sensitivity of video to packet loss, increased care needs to be taken when it is streamed over a wireless network.

WLANs can be categorised into two modes, Ad-hoc and Infrastructure. In an Ad-hock network there is no centralised infrastructure and each node within the network operates as a peer. In an Infrastructure network, access points (APs) are usually connected to a LAN through which the wireless clients will have access to Internet services. In this thesis we are interested in the streaming of video and will concentrate our efforts on the Infrastructure type wireless networks.

3.2.1 802.11

The IEEE 802.11 standard provides two different MAC functions to coordinate different wireless devices. These are the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). DCF provides a distributed means for devices within the network to coordinate the sending of packets, with the aim that packets should not collide within the wireless medium. However, there are no guarantees as all nodes contend for the medium equally. DCF only works where wireless nodes are able to overhear other wireless nodes communicating, but this is not always the case. PCF is provided as an alternative to DCF. PCF enables the access point to coordinate access to the medium and provides contention-free access to the medium.

To ensure the interoperability of 802.11 devices from different manufactures, the Wi-Fi Alliance was formed to certify compliance. To receive certification wireless devices only have to implement the DCF function and, consequently, PCF is rarely implemented. Due to the widespread use of DCF it is discussed in greater detail below.

3.2.1.1 Distributed Coordination Function

DCF implements carrier sense multiple access with collision avoidance (CSMA/CA). With CSMA/CA a station wishing to transmit must sense the channel first and if another station is transmitting then it must wait before transmitting into the channel.

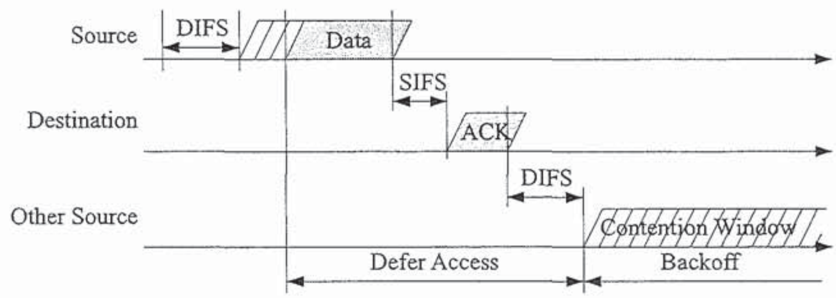


Figure 3.11: DCF basic access example [7]

Figure 3.11 shows how DCF operates. When a packet is to be put onto the network the station initially senses the medium for the duration of one DCF Interframe Space (DIFS). If the medium is busy then the station waits until the medium becomes free again. Once the medium is free there are two inter frame periods to be used. One is Short Interframe Space (SIFS), the duration after the medium becomes free, during which an acknowledgement (ACK) can be returned from the recipient to the sending node. The other is the DIFS. Once these two periods have passed without anyone using the medium, the transmitter senses the medium during the contention window.

The reason for the contention window is, while a station has been transmitting it is likely that multiple other users have decided to access the medium. In this situation, a number of stations will likely have been listening for the end of the transmission and if they all start transmitting at the same time then all the transmissions will collide and none of them will be received by the destination.

The contention window operates by selecting a random value in the range $[0, CW-1]$, termed the back off value. When a station first enters the contention window $CW = CW_{min}$. The station decrements the back off value for each time slot where the medium remains free. When the back off value gets to 0 and if the medium is still free then the station may start transmission. Since the back off value is selected at random it is possible that another station will have selected a lower back off value and will start transmitting earlier.

Once the node has transmitted its frame it waits for an ACK from the destination. If the ACK is received then the station has successfully transferred the packet and is able to start the process again to transmit another packet. If the ACK is not received within a time-out period then the retry counter is incremented and the back off procedure is repeated, but with a new value of $CW = 2 \times CW$ up to $CW = CW_{MAX}$. If the retry counter reaches the retry limit, usually 7, the packet is dropped and the process restarts to transmit another packet.

3.2.2 802.11e

The original IEEE 802.11 standard provided no QoS mechanism. IEEE 802.11e implements QoS mechanisms through the addition of the Hybrid Coordination Function (HCF).

3.2.2.1 Hybrid Coordination Function

HCF takes both DCF and PCF from the 802.11 standard and enhances them with QoS-Specific mechanisms. As in 802.11, both a contention-bases access method, termed enhanced distributed channel access (EDCA), and a contention-free access method, termed HCF controlled channel access (HCCA), are defined in the 802.11e standard.

As with the DCF and PCF case the Wi-Fi Alliance only requires contention-based access (i.e. EDCA) to be implemented for certification, as such HCCA is rarely implemented. The 802.11e certification is named Wireless Multimedia Extensions (WME) or Wi-Fi Multimedia (WMM).

3.2.2.2 Enhanced Distributed Channel Access

EDCA takes the eight 802.1D user priority classes and assigns them to four EDCA ACs, as shown in Table 3.1. EDCA differentiates between different classes by changing four parameters. We group these changes as Contention Window Control (containing two parameters),

Table 3.1: 802.1D user priority to AC mapping [2]

Priority	802.1D [105] user priority	802.11e [2] AC	Designation
Lowest	1	AC_BK	Background
	2	AC_BK	Background
	0	AC_BE	Best Effort
	3	AC_BE	Best Effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
Highest	7	AC_VO	Voice

Arbitrary Inter Frame Spacing (AIFS) and Transmission Opportunities (TxOP). A typical set of parameters is shown in Table 3.2.

Table 3.2: Sample Class Parameters

Access Class	Priority	CW_{min}	CW_{max}	AIFS	TxOP
AC_BK	Lowest	15	1023	7	0
AC_BE	Low	15	1023	3	0
AC_VI	High	7	15	2	3008 μs
AC_VO	Highest	3	7	2	1504 μs

Contention window control EDCA makes changes to the contention window value from DCF. Instead of being randomly selected in the range $[0, CW-1]$, EDCA changes this to a range of $[CW_{min,class}, CW-1]$. Additionally, the CW_{max} value is changed to become class specific, i.e. $CW_{max,class}$.

The benefits of giving more important classes lower $CW_{min,class}$ and $CW_{max,class}$ values are that the contention window value will expire earlier than those with larger $CW_{min,class}$ and $CW_{max,class}$ values. When the contention window expires and if no other station is transmitting then the station will be able to transmit its packet.

Figure 3.12 shows a simplified example where three stations each have a packet of a different access class to transmit. The figure shows the effect that the different contention window values have on the order the packets are to be sent. Using the typical set of parameters shown in Table 3.2 the contention windows of 5, 10 and 20 are randomly chosen for AC_VO, AC_VI and AC_BE, respectively.

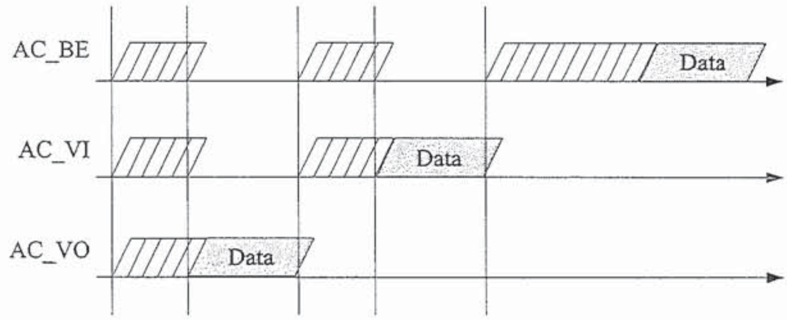


Figure 3.12: EDCA Contention Window Example

Arbitrary inter frame spacing Within the DCF standard SIFS and DIFS are used to provide two levels of priority for ACKs and data, respectively. This allows ACKs to have access to the medium before another station starts their contention window countdown. In a similar manner, Arbitrary Inter Frame Spacing (AIFS) replaces DIFS and provides each AC with its own AIFS duration. This allows higher priority classes to contend for the medium earlier than lower priority classes. Figure 3.13 shows the effect different AIFSs have on the start of the contention for the media.

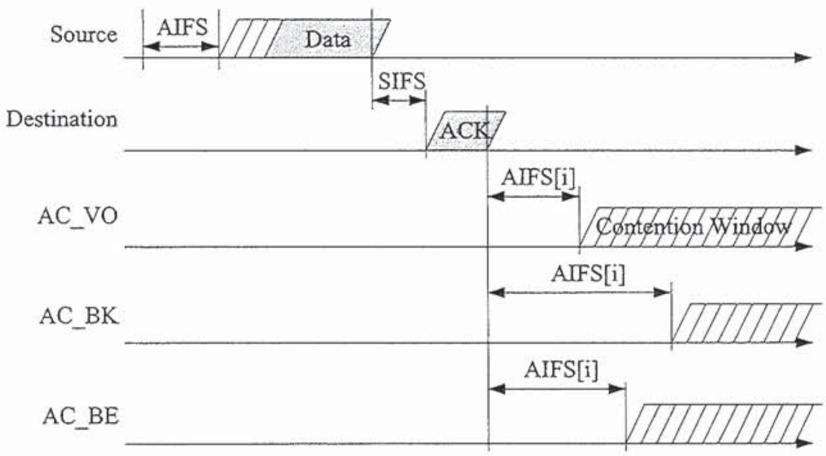


Figure 3.13: EDCA access example

It might appear that the AIFS could just be replaced with a larger CW_{min} value, however with the use of AIFS the higher priority classes can start transmitting before the AIFS of lower priority classes has expired. Stations which put their traffic in the higher priority classes can starve stations with lower priority classes from accessing the medium, which would not be the case with changes to the CW_{min} value.

Figure 3.14 demonstrates how the use of different AIFS values can starve lower importance classes from access to the medium. Figure 3.15 shows the same scenario as in Figure 3.14 but with constant AIFS values. The difference between figures 3.15 and 3.14 highlights the care which needs to be taken in the choosing of AIFS values [106].

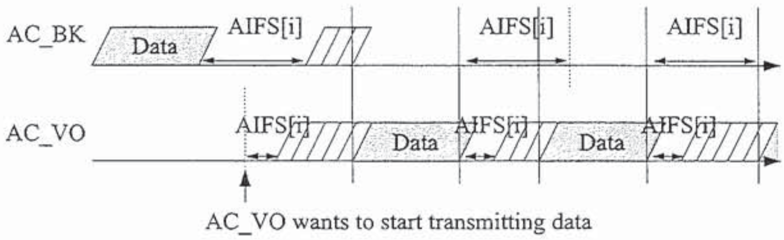


Figure 3.14: AIFS access example

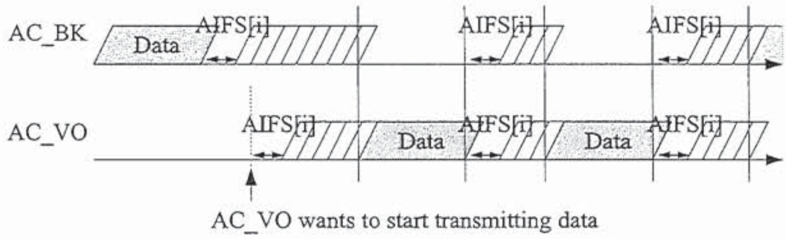


Figure 3.15: Constant AIFS access example

Transmission opportunities Transmission Opportunities allow a station to transmit multiple packets without having to re-contend for the network. This is useful in both the voice and video cases where an audio segment or video frame might not fit into a single packet.

3.2.2.3 Access Class Mapping

802.11e access class is a link layer example of DiffServe, as previously discussed in Subsection 3.1.2. Through the inspection of the DSCP field within the IP packet header the packet is assigned to one of the access classes. Currently, there is no clear assignment between DSCP values and ACs, which is left to individual manufacturers or network administrators to decide [68].

3.3 Conclusion

Of the existing QoS frameworks, IntServ provides guarantees over QoS, however this is not scalable to an Internet size network. The alternative framework, DiffServ, provides no service guarantees and is unable to prevent issues resulting from router updates. As it has been seen, BGP updates can temporarily (an average of three but up to fifteen minutes [75, 79, 80]) cause increased packet disturbance. We look at network diversity as a potential technique to reduce bursts of packet losses.

Of the presented techniques, Source Routing would provide the best method of utilising diversity, since it would require no additional hardware or protocol adaptation. The reason that Source Routing is not chosen is because ISPs actively filter out packets which use source routing to prevent the security risks which they pose. We also discount router alteration as this would require large investments in expensive equipment for the diversity to be exploited.

Of the remaining technologies, such as multihoming and overlay networks, the benefits of the two are comparable. The decision between overlay networks or mutlihoming is likely to be based on cost and development issues as opposed to technical reasons [107]. Of these two, overlay networks are likely to be the cheaper one as it only requires infrastructure to a single ISP and a single contract, whereas multihoming requires contracts with multiple ISPs and the related additional infrastructure.

Finally, wireless QoS is discussed. The recent IEEE 802.11e standard has introduced class based QoS to IEEE 802.11 networks. Prior to IEEE 802.11e all packets transported across the wireless link were treated equally. Since the introduction of 802.11e access points now permit four different access classes to differentiate between different types of traffic.

Chapter 4

Loss Performance of Overlay Networks

In this chapter, we look at one QoS related measurement, packet loss, and investigate if this can be improved through the use of path diversity. We develop previous authors' work [94–96] by extending the number of paths from two, up to eleven. Our initial work establishes simulation models for three different packet loss scenarios. The results presented demonstrate how the different loss scenarios affect the packet loss patterns when path diversity is used.

The simulation results are supported with analytical models, initially for each lossy environment and finally a combination of them. The reason for presenting a combined model is that in a real network losses are likely jointly contributed from different sources.

The simulation and analytical models show two potential trends when using path diversity, firstly increasing the number of paths increases the proportion of packets lost for both the burst and random loss environments and secondly increasing the number of paths decreases the number of packets lost in a congestion environment. For the combined model, there is the possibility that an optimal number of path can be chosen, maximising the benefits from using path diversity.

To investigate if the Internet shows a clear trend we take results from a real overlay network implemented on PlanetLab. From this we show how the use of real overlay networks effect the packet loss patterns received by the client.

4.1 Packet Loss Models

In any network the causes of packet losses can be classified into three main types: Random losses, caused by poor channel conditions; Burst losses, mainly due to link or node failure in wired systems or severe channel destruction such as fading in wireless systems; and Congestion, due to limited buffer, bandwidth and processing capability at network routers. In reality, however, when measuring the losses that are experienced by a stream it is difficult to separate the losses into their root cause, random, burst or congestion. However, estimated parameters can be made available in most cases, especially for random and burst losses. Given an overlay network, as shown in Figure 4.1, we split the stream equally between each path involved in order to simplify the resource management, resulting in the ability to implement path diversity without a reverse channel (enabling feedback from the client to be returned to the server), should this be desired.

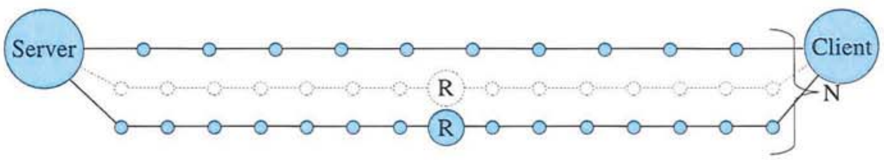


Figure 4.1: Sample topology with three paths

The splitting of the original stream into multiple sub-streams does not add any additional overhead since no additional packets are sent nor are packets split, which would require additional header and routing overheads. The topology shown in Figure 4.1 would be an ideal overlay network topology. In our work we initially show the effect which the overlay network has on this ideal topology after which we look at less ideal but more realistic topologies including common links.

4.1.1 Random Losses

Transmitting packets over an individual link can be described using a Bernoulli trial or model with packet loss probability p and loss-free probability $1-p$. We can extend the Bernoulli model to a system consisting of N paths with each of them having L_n links, where $n = 1, \ldots, N$. Let p_i be the packet loss probability of the i-th link in the system, where $i = 1, \ldots, \sum_{n=1}^{N} L_n$.

For the individual links of a given path, we assume independent packet losses defined by the Bernoulli model. Hence the packet loss probability of the n-th path in a system using UDP, where no retransmission of a lost packet is available, is given by

$$P_n = 1 - \prod_{i=J+1}^{J+L_n} (1 - p_i) \tag{4.1}$$

Where J is the total number of the links of all the previous $(n-1)$ paths, i.e. $J = \sum_{j=1}^{n-1} L_j$. If the loss probabilities of all the links in the system are assumed to be the same, i.e., $p_0 \ldots p_{\sum_{n=1}^{N} L_n} = p_R$, then Equation (4.1) can then be simplified to

$$P_n = 1 - (1 - p_R)^{L_n} \tag{4.2}$$

We also assume that packets are distributed evenly between different paths. The packet loss probability for random losses of the N-path system, $P_{R,N}$, can therefore be expressed by

$$P_{R,N} = 1 - \frac{1}{N} \sum_{n=1}^{N} (1 - p_R)^{L_n} \tag{4.3}$$

As a quick example by using Equation (4.3) in the traditional client-server environment, we look at a single path system with $p_R = 0.0001$, $N = 1$ and $L_1 = 12$, which results in a packet loss probability $P_{R,1} = 0.0012$. In the multipath cases, the primary path between the server (the path which does not use a relay node as shown in Figure 4.1) and the client is normally the shortest path of all possible ones. In other words, it is most likely that the alternative paths have more links than the primary path. Following the above example, we include two additional paths to the first one, which makes $N = 3$. If we set $L_2 = L_3 = 16$, then $P_{R,3} = 0.0015$ according to Equation (4.3). It is observed from these two examples that when the number of links on the additional paths are greater than that for the primary path, and if all links experience the same loss rate, then increasing the number of paths leads to an increase in the packet loss probability of the system.

4.1.2 Burst Losses

The second cause of loss is from burst losses. We describe burst losses on a single link as a Markov chain, as depicted in Figure 4.2, with two states: good and bad (where packets are dropped), and the corresponding transition probabilities: p_t and q_t.

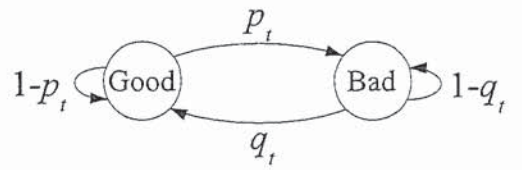


Figure 4.2: Two state Markov chain, burst loss model for single link

Consider a time invariant Markov chain. The probability transition matrix of the model is given by

$$M_p = \begin{pmatrix} 1-p_t & p_t \\ q_t & 1-q_t \end{pmatrix} \tag{4.4}$$

Let p_1 and p_2 be the probabilities of being in the good and the bad state, respectively, and satisfy the condition of stationary distribution, i.e., $\begin{bmatrix} p_1 & p_2 \end{bmatrix} \times M_p = \begin{bmatrix} p_1 & p_2 \end{bmatrix}$. The probability of being in the bad state, which is equivalent to the packet loss probability for burst losses of a single link, $p_{B,N}$, can be calculated from the following equation with the condition $p_1 + p_2 = 1$, i.e. the system must always be in one of the states. The probability of being in the burst state can be calculated from

$$p_2 = p_{B,N} = \frac{p_t}{p_t + q_t} \tag{4.5}$$

Unlike p_R in the random loss case, $p_{B,N}$ has dependency on the number of paths, N, in the multipath dissemination environment. If again we assume that all N paths with L_n links, where $n = 1, \ldots, N$, have an equal opportunity to deliver packets, then the packet loss probability for burst losses in an N-path system, $P_{B,N}$, can be expressed by

$$P_{B,N} = 1 - \frac{1}{N} \sum_{n=1}^{N} (1 - p_{B,N})^{L_n} \tag{4.6}$$

4.1.3 Congestion

To model congestion we simplify network conditions into two states, congested and non-congested. In the non-congested state all the packets from the source are able to reach the destination without loss. In the congested state the number of packets lost due to congestion depends on various factors. In our model we assume a first-in-first-out(FIFO) drop tail queue, and that any packet phase effects that might exist are short lived because of the random pauses during the transmission of background packets [108].

Let R_I be the average total packet input rate at a node, as illustrated in Figure 4.3, which is a combination of our monitored stream with input rate, $R_{S,N}$, as well as the background traffic input rate, R_B, i.e., $R_I = R_{S,N} + R_B$. $R_{S,N}$ relates to the total number of paths (N) involved, but R_B is static, thus independent of N. For a given period of observation time, we may divide it into two time slots: T_1 when only the streamed packets enter into the node ($R_B = 0$), referring to the non-congested state; and T_2 when both the streamed and background traffic enter into the node ($R_B \neq 0$) , referring to the congested state.

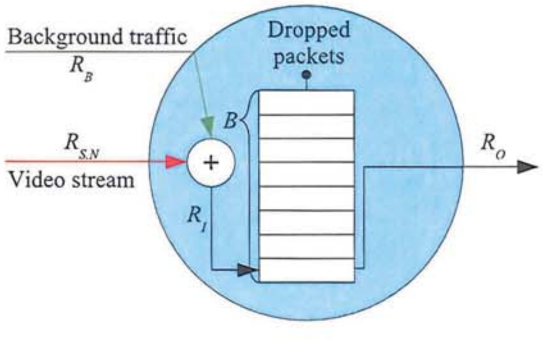


Figure 4.3: Link congestion model

Let R_O be the maximum packet output rate of the node with a buffer of size B packets available. The packet loss probability due to congestion for a single node is given by

$$p_{C,N} = \frac{B_D}{R_{S,N} \times (T_1 + T_2) + R_B \times T_2} \tag{4.7}$$

where B_D is the number of packets lost during the period of observation. B_D is meaningful only when $R_I > R_O$ and $(R_I - R_O)T_2 > B$ and is determined by

$$B_D = (R_I - R_O) \times T_2 - B$$

With the same approach as before, the packet loss probability for congestion losses in the N-path system, $P_{C,N}$, can be expressed by

$$P_{C,N} = 1 - \frac{1}{N} \sum_{n=1}^{N} (1 - p_{C,N})^{L_n} \tag{4.8}$$

4.2 System Models and Simulations

To simulate multipath dissemination, an overlay network is implemented in J-Sim [109]. Our initial work assumes that the server and clients are multihomed, allowing them to have a number of physical connections which in our work are connected to a network with many redundant paths. Our model is made up of one server and one client with N paths in between.

To form an overlay network relay nodes are used, which work at the application layer to forward packets on each path except for the primary path. An example of this network is shown in Figure 4.1. Each path consists of a number of links, with each link having a bandwidth of 2Mb/s and a propagation delay of 4 milliseconds, which is the same as in [110]. The total streamed data rate is 800kbps and split equally for each path.

We assign packets to the paths in a round robin manner. In some systems [94], a packet is divided into N smaller packets, which are then sent down N paths (one-to-one) in parallel. The amount of information transferred is the same for both dissemination algorithms. The disadvantage of splitting the packets is that with each extra packet transferred there are additional requirements such as buffering and routing in the network [111].

4.2.1 Path Length

Some authors fix the number of links for both the primary and secondary path to just a single number of links for the primary path and a single value for all other paths in their simulations. In the case of [110] the primary path consists of 11 links and all other paths are 18 links long. In our model, however, we set out to find a more suitable set of lengths for the additional paths, as in reality the length of these paths will increase as more paths are added. To find these lengths we created over 1,000 two-layer topologies to simulate the Internet [112, 113].

The first of the two layers is to represent AS interconnections, the second layer is to represent internal AS routers. The AS inter connections are modelled using an AS Waxman model; and the internal connections are modelled using a Barabasi-Albert2 model. In the model we connected 75 ASs together, each assumed to have 20 internal routers. The Bright configuration file is included in Appendix B.

Using the generated topologies a server and client are randomly selected from all of the available hosts, and then from the remaining nodes a further 10% are selected to act as relays. The number of links for each path can now be calculated based on the topologies generated and the locations of the relay nodes. Using this data we are able to list the paths in terms of their length and select the shortest one as the primary path. Figure 4.4 shows the number of links for each path calculated as the average of over 1,000 random two-layer topologies. For simulation purposes, the values are rounded to the nearest number of links.

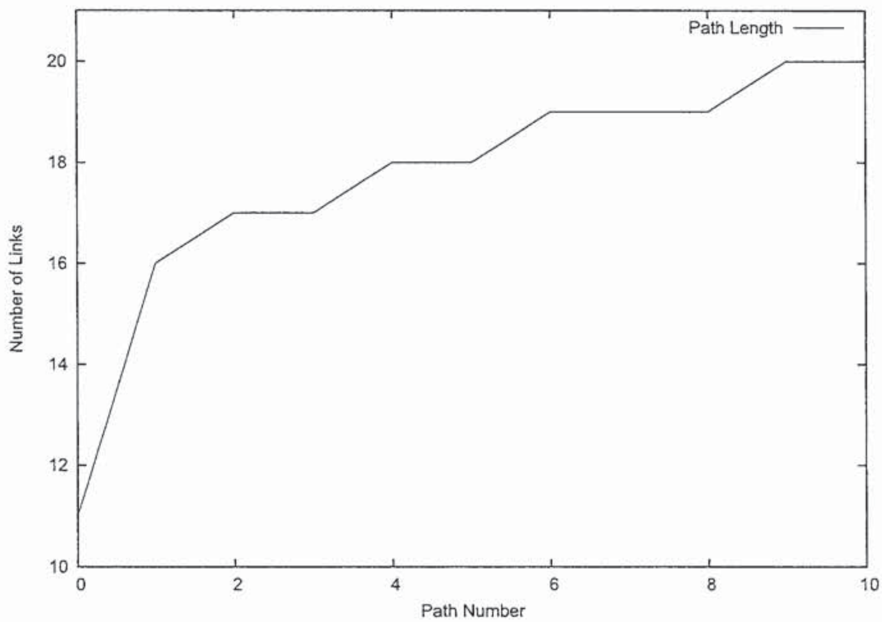


Figure 4.4: Number of links on each path

4.2.2 Simulator

The majority of previous work regarding path diversity has been conducted using simulation. Simulation enables a controlled environment on which we can examine the effect of path diversity.

There are many different simulation packages available, such as Network Simulator version 2 (NS-2) [114, 115], Modeler [116] and J-Sim [109]. Each simulator has different strengths, which are briefly reviewed in relation to the simulation of path diversity.

NS-2 is an open source packet level simulator, which is widely used within the research community. Using a combination of the Tool Command Language (TCL) and C++ NS-2 allows both quick prototyping and high speed processing. NS-2 has the disadvantage of the loose combination of TCL and C++ means that finding existing code is difficult. It is possible for the aspects of simulation to be written in TCL or C++, meaning that debugging code can prove difficult especially when part of the implementation of a class is in both TCL and C++.

Modeler is a closed source simulator with a larger library of components. Although implementing new applications within Modeler is possible, the lack of transparency of the source code makes it difficult to add extra instrumentation when this would help to identify what is actually happening.

J-Sim is an autonomous component based open source simulator. An autonomous component architecture (ACA) allows a component to have multiple inputs and multiple outputs. The inputs and outputs of multiple components are then connected together. When the simulator runs, data is then passed from output pins of one component and are passed to inputs of another. A component itself can be made up of a number of connected sub-components. Although J-Sim is a fairly generic simulator, an extensive package of packet switching network modelling tools have been developed for it.

The advantage J-Sim has over both Modeler and Ns-2 is that standard java applications can be run inside the simulation environment. Due to the modularity of java the TCP/IP stack of an application can be replaced with a component which then interfaces with the simulator. This enables applications to be written for a simulation environment and then to be taken and used in the real world. The advantage of this is a reduction in the amount of code and debugging which needs to occur. Additionally, this prevents two implementations of the code from not functioning identically.

J-Sim is written in java, but uses TCL, java or python to set-up simulations [117]. The reason that J-Sim does not experience the same split language problem as Ns-2 is because all components of the simulator are required to be compiled into the java code, with TCL, python or additional java code only left to set up the inter object connections.

Through the rest of the simulation work J-Sim is chosen. This is because of its simple programing language split and with the ability to run real java applications thus removing to write applications twice for simulation and for execution. Although J-Sim interfaces with the java socket and server socket classes, for use with TCP, this was not the case for the User Datagram Protocol (UDP). Therefore, J-Sim has to be developed to simulate the packet types that we want to use.

The simulations are used to identify if the use of path diversity can reduce the loss rate, and if so under which network conditions. We only assume a packet switched network, so in reality this could be wired or wireless. We establish three loss models based on the causes of loss, congestion, random and burst losses, which lead to the creation of a combined loss model to be discussed in Subsection 4.3.

4.2.3 Random Losses

By assuming that it is always possible to create N diverse paths between a client and a server, we can simulate the effect of random losses. The model requires only one parameter: the packet loss probability of individual links, p_R . For the Internet environment we use $p_R = 0.002174$ calculated based on a path length of 11 using the values presented by the authors of [118].

Figure 4.5 shows the average probability of a packet being lost from 10 simulations using different random seed values, with each being simulated for 1000 seconds. In addition to the simulation values, analytical values calculated from Equation (4.3) are presented for comparison. From this we can see that the packet loss probability increases as the number of paths increases. This is in line with the analytical results presented.

4.2.4 Burst Losses

To simulate burst losses we used the probability transition matrix, given in Equation (4.4). For the simulations to be as accurate as possible we would need to evaluate the matrix at a very high frequency, but this proves to be very difficult to find existing values from the research community. A solution to this problem is to reduce the number of evaluations. Though the accuracy of the results will be affected to a certain degree we deem that they are still appropriate for validating the model. We average out the unicast values [118] to gain the parameters $p_t = 0.0014$ and $q_t = 0.854$. The loss model state, good or bad, is re-evaluated every 0.06 seconds.

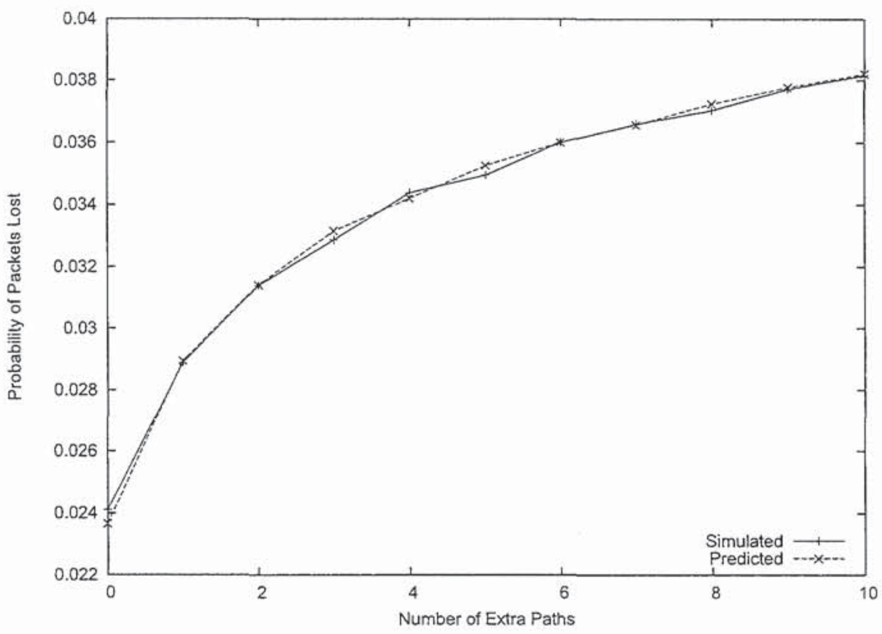


Figure 4.5: Packet loss probability for the random loss model

Figure 4.6 shows the packet loss performance. The results from Equation (4.6) are included for comparison. We see that the simulation and analytical results are both very close. As the number of paths increases the loss probability increases as well. The trend shown is very similar to the trend shown for random losses in Figure 4.5.

4.2.5 Congestion

To cause packet loss, we included exponential on/off traffic down each link. We configure the traffic so that on average after 8 seconds without any background traffic is then followed by 40 milliseconds of traffic, with a rate of 2Mb/s. Each node is set to have enough space to buffer just one packet. Since each network link is limited to 2Mb/s, when the background and video source try to flow down the same link there will be packet loss once the buffer overflows.

Figure 4.7 shows that increasing the number of paths reduces the packet loss probability. The improvements gained in the congested environment are through utilising available buffer space on the additional paths. This would not only reduce the losses received in the stream but also in any other streams which are sharing the same paths.

The problem with using multiple paths is that not all systems are multihomed, especially

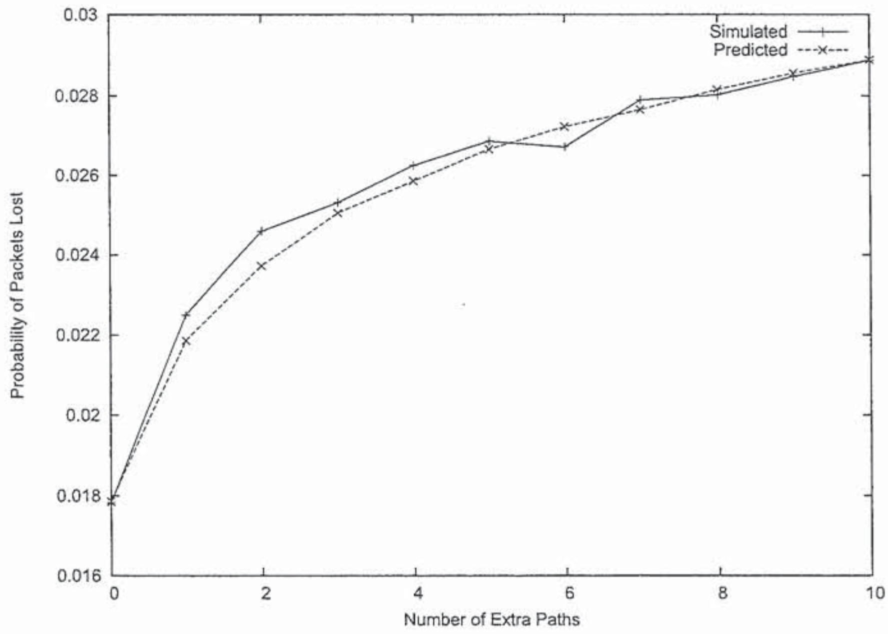


Figure 4.6: Packet loss probability for the burst loss model

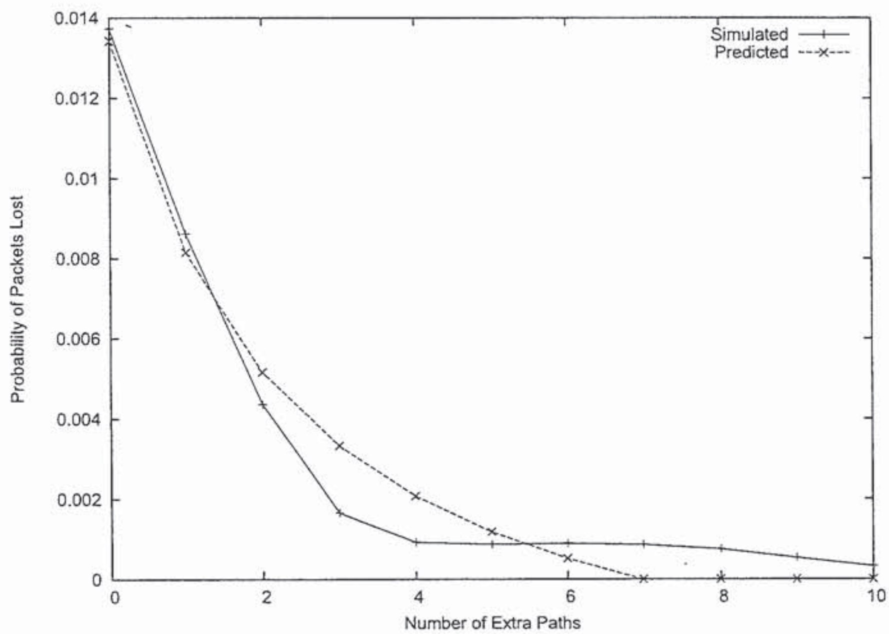


Figure 4.7: Packet loss probability for the congestion model

in a residential environment. Combined with general Internet routing it is likely that some common links will exist. To investigate the effect of common links we simulated our network with the same traffic and created a network topology containing two additional paths (three in total) with varied numbers of common links at the source and at the destination.

Figure 4.8 shows that the number of common links affects the performance of path diversity. Therefore to achieve the maximum benefit the paths need to be as diverse as possible. Additionally, we note that there is little difference between having common links at the source or at the destination.

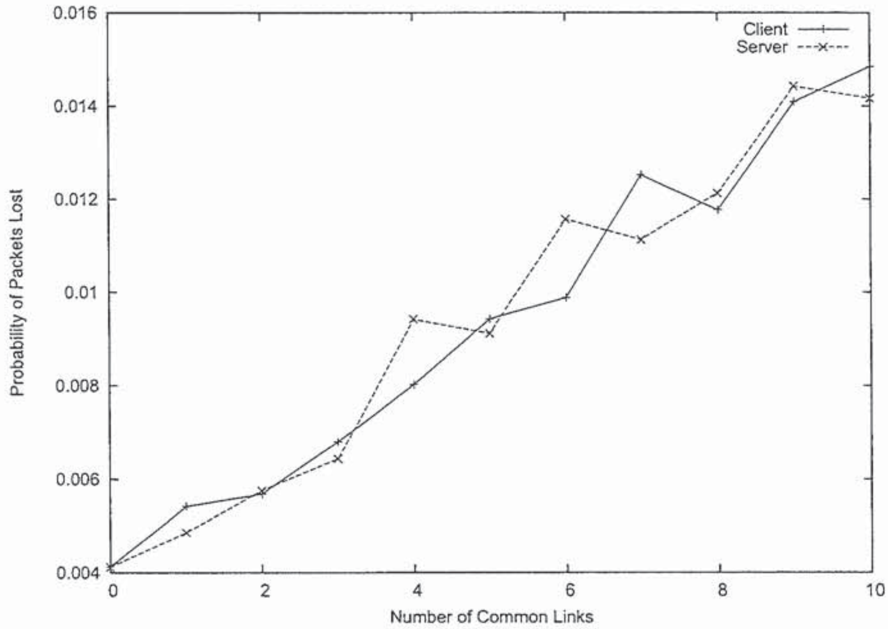


Figure 4.8: Packet loss probability for system with common links near either the server or client

4.3 Combined Loss Environment

From the three loss scenarios presented, it is shown that path diversity can improve the packet loss rate in the case of congestion. This is a useful result when deploying multihoming or overlay networks. However, in a real environment losses are contributed by various sources. To evaluate the effect of this environment, we combine the three individual loss models presented previously into a single model.

4.3.1 Combined Model

Figure 4.9 depicts a graphical representation of the proposed combined model, which operates at each node. The combined loss model is created by passing the packets, sequentially, through each loss model. The packet loss probability for each section (one node and one link) can be calculated from the following calculation

$$p_{A,N} = 1 - (1 - p_{C,N}) \times (1 - p_{B,N}) \times (1 - p_R)$$

where $p_{C,N}$ is defined in Equation (4.7), $p_{B,N}$ is defined in Equation (4.5) and p_R as used in Equation (4.2).

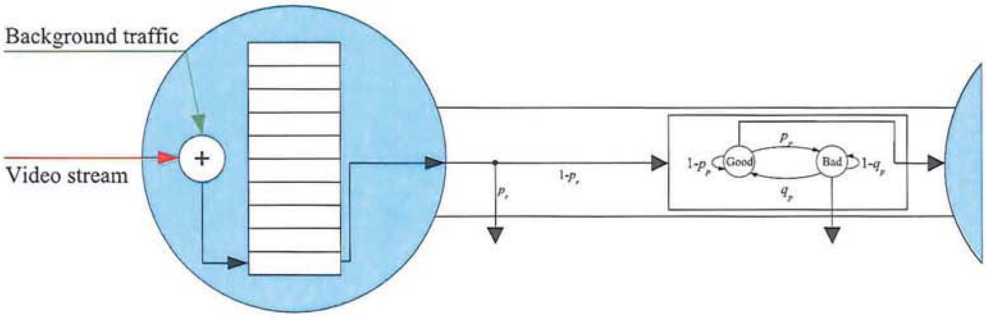


Figure 4.9: Combined model

Using the same approach as in Subsection 4.1, the packet loss probability for the N-path system can be expressed by

$$P_{A,N} = 1 - \frac{1}{N} \sum_{n=1}^{N} (1 - p_{A,N})^{L_n}$$

4.3.2 Combined Model Results

The problem with a combined loss model is that there are more parameters to set up than in the individual model case. In our simulations we use the following settings: $p_t = 0.001$, $q_t = 0.854$, $p_R = 0.001174$, and the settings for the congestion model are the same as in Subsection 4.2.5.

The results shown in Figure 4.10 indicate that in a combined loss environment the system can still benefit from the use of multipath dissemination. However, increasing the number of paths is not always beneficial, as Figure 4.10 demonstrates. The reason that increasing the number of paths is not always beneficial is because adding additional paths are normally longer and as seen in the random and burst cases are lossier since the shorter paths with the lower packet loss rate are selected first. When the benefits received by the reduction in congestion does not exceed the additional packets lost which result from the addition of longer paths then path diversity no longer proves to be beneficial. Nevertheless, our results suggest that an optimal number of paths for the given conditions could be obtained for minimising the overall packet loss rate. However, this optimal number of paths could change as the network characteristics vary.

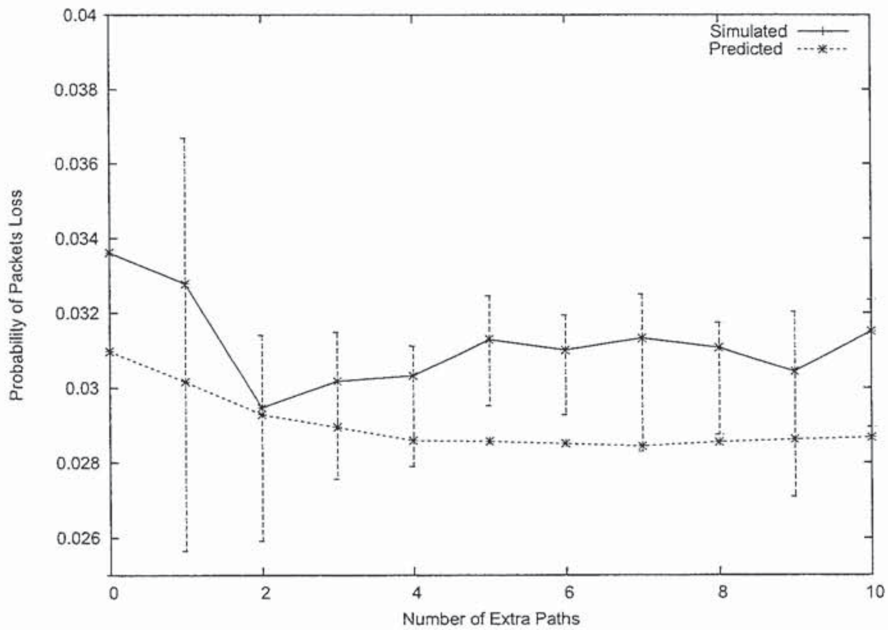


Figure 4.10: Packet loss probability for the combined model

In consumer networks the last link to an individual's home can be of a different media, although high speed fibre optic links is the normal choice for the Internet backbone. These last links, such as IEEE 802.11, ADSL or fibre, can have different properties from the backbone. The wireless medium, for example, has a higher loss rate than the wired links, and although these losses can be reduced by retransmission at the data link layer its performance characteristics are still different from a wired last link. Our loss models to be applied to this scenario needs to be modified slightly as a result. Let $p_{A,N,B}$ be the error probability for the backbone of the network, $p_{A,N,L}$ the error probability for the last link, $p_{B,C,N}$, $p_{B,B,N}$ and $p_{B,R}$ the associated congestion, burst and random loss probabilities on the backbone, respectively, and $p_{L,C,N}$, $p_{L,B,N}$ and $p_{L,R}$ the associated congestion, burst and random loss probabilities on the last link, respectively. Then we have

$$p_{A,N,B} = 1-(1-p_{B,C,N})\times(1-p_{B,B,N})\times(1-p_{B,R})$$
$$p_{A,N,L} = 1-(1-p_{L,C,N})\times(1-p_{L,B,N})\times(1-p_{B,R})$$

Using the same approach as above, the packet loss probability for the N-path system can be expressed by

$$P_{A,N} = 1-\frac{1}{N}\left(\sum_{n=1}^{N}(1-p_{A,N,B})^{L_n-1}\right)\times(1-p_{A,N,L})$$

We simplify our models with the assumption that random packet loss is just a special case of burst packet loss, where the burst length is a single packet. Using this assumption then we can classify each section (the backbone links or the last link) into one of two possible states: either experiencing congestion or burst losses. This will result in four scenarios: burst-burst, congestion-congestion, congestion-burst and burst-congestion.

The first two scenarios are the simplest, as they follow the same trend as if the last link was the same as the backbone, which have been evaluated in Subsections 4.2.4 and 4.2.5. Figure 4.11 shows how the probability of the last link affects the overall probability of packet loss in the burst-burst scenario.

For the congestion-burst scenario, Figure 4.12 shows that when the last link experiences random losses while the rest of the network experiences congestion, such as in the case of wired-wireless networks, multipath transmission can still improve the loss rate of the system.

In addition, path diversity can also produce benefits by inherently dispersing losses across different paths, which provides opportunities for other techniques such as forward error correction (FEC). As FEC codes can only control a certain number of losses, it is important to know the loss property of the paths in order to design a proper code for a certain application.

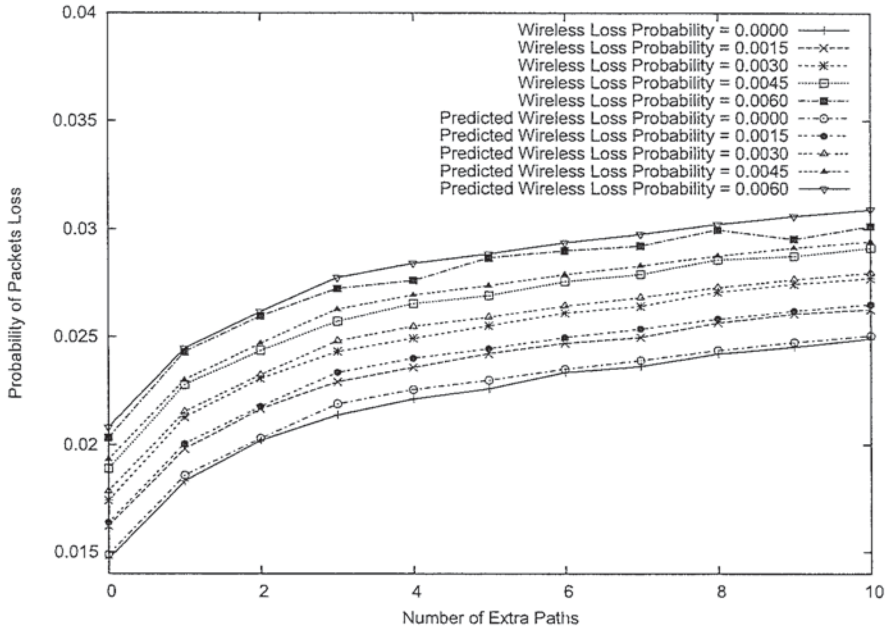


Figure 4.11: The effect of altering the burst probability on the last link in the burst-burst scenario.

In our work, we form groups of 30 packets to examine the maximum number of losses in the groups. Figure 4.13 shows this feature for all three packet loss models. It can be seen that increasing the number of paths dramatically reduces the maximum number of errors per group in burst and congestion models.

4.4 Experimentation

We have previously highlighted that path diversity may not be able to reduce the packet loss rate in all scenarios. We found that, however, in predominantly congested scenarios (as opposed to ones experiencing random or burst losses) path diversity can produce benefits. Therefore, we set out to investigate if the use of path diversity can produce any benefits over the Internet. For this purpose we created overlay networks using the PlanetLab testbed on the Internet. It has always been assumed that the wired Internet experiences mainly congestion losses [119] and fewer non-congestion losses of data packets. This assumption has been used in the fundamental design of many protocols. In our investigation we want to test the losses from the two different sources by using UDP, because it has no error recovery methods.

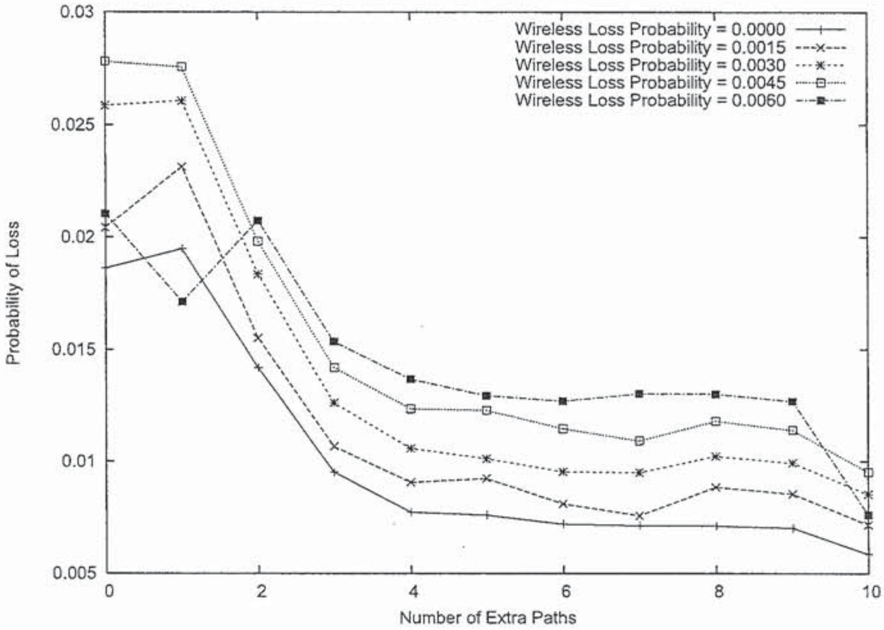


Figure 4.12: The effect of altering the burst probability in the congestion-burst scenario.

4.4.1 PlanetLab Setup

In our investigation of path diversity we set up a number of relay nodes in different ASs to create an overlay network. The reason for having relays is to force the IP packets to be routed along paths which they would not normally take to reach their ultimate destination.

Using the PlanetLab nodes shown in Table 4.1 we deployed our software to create an overlay network. The path diversity software consists of three parts; a server, relay, and sink.

The server generates packets at a regular intervals (in our case every 20ms), which are then transmitted over the network. Packets are sent in turn to the sink directly or to different relay nodes in a round robin manner, for each packet in turn. The content of the packet is an identifying number with padding to make the payload up to 1000 bytes long which should not be fragmented by the network as it is smaller than the Ethernet maximum transmission unit (MTU) which is typically the major cause of fragmentation.

When the experiment starts the relay node is sent details on the destination address and port number. All the packets that arrive at the relay application are then forwarded to the destination. The sink node logs the identifier numbers of the received packets and the local time of its arrival. Once the experiments have been completed the resulting log files are

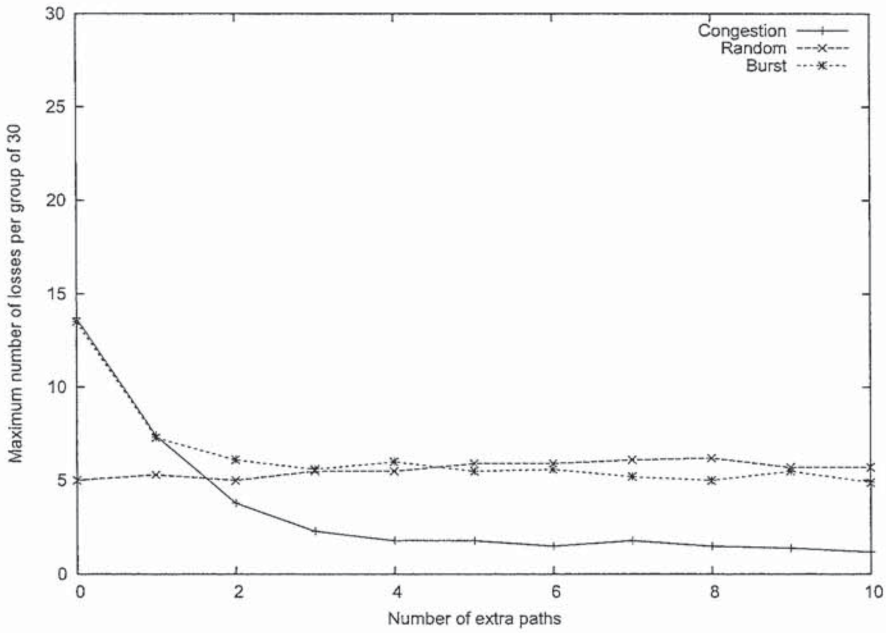


Figure 4.13: The maximum number of losses in a group of 30 packets in different error environments

processed.

To collect a range of results we completed the same type of experiments with three different groups of PlanetLab nodes. At most one node was selected from a PlanetLab site and each node selected would only participate in one group to minimise any interference that might be caused between them. For each group of nodes, the two most geographically dispersed nodes were selected, one to be the source node and the other the sink node. The remaining nodes were assigned as relay nodes. The length of the path from the source to each relay node was computed using traceroute [120] as well as the number of links from each relay node to the sink. The sequence of relay nodes was selected to minimise the total number of links making up one path. The path lengths in terms of the number of links are shown in Figure 4.14. Figures 4.15, 4.16 and 4.17 show the overlay topologies of our experiment for each of our three overlay networks, US1, US2 and EU respectively.

Starting with just the primary path we streamed packets over the Internet for a period of 24 hours. We increased the number of paths used every 24 hours by adding a relay node until all the nodes in a group were utilised. 24 hours after all the nodes were added the experiment would restart. As nodes can be disconnected at any time and the overlay topology needs

Group	Task	Nodes
US1	Source	righthand.eecs.harvard.edu
	Destination	planet2.cs.ucsb.edu
	Relay 1	PlanetLab2.cs.uoregon.edu
	Relay 2	PlanetLab2.csee.usf.edu
	Relay 3	pl2.cs.utk.edu
	Relay 4	PlanetLab2.unl.edu
	Relay 5	PlanetLab2.cse.msu.edu
US2	Source	planet-lab1.cs.ucr.edu
	Destination	planet2.scs.cs.nyu.edu
	Relay 1	PlanetLab1.utep.edu
	Relay 2	PlanetLab1.cs.uchicago.edu
	Relay 3	PlanetLab2.cs.purdue.edu
	Relay 4	planetslug2.cse.ucsc.edu
	Relay 5	plab2.eece.ksu.edu
EU	Source	PlanetLab2.sics.se
	Destination	PlanetLab1.fct.ualg.pt
	Relay 1	PlanetLab2.ls.fi.upm.es
	Relay 2	PlanetLab02.mpi-sws.mpg.de
	Relay 3	PlanetLab2.olsztyn.rd.tp.pl
	Relay 4	PlanetLab2.nrl.dcs.qmul.ac.uk
	Relay 5	PlanetLab1.cs.uit.no

Table 4.1: Grouped Nodes

to be kept constant, we only examined the overlay networks where all the relay nodes are present. When a node was disconnected then only the experiments not requiring that node were examined. The complete experiment was conducted over a period lasting 50 days which resulted in every group being tested at least 3 times with some paths being examined 11 times.

4.4.2 Experiment Results and Analysis

The results presented here are always the mean values from the processed data with one standard deviation plotted. Figure 4.18 shows for all three groups how the use of path diversity affects the total packet loss rate. In the worst case the use of path diversity increases the average loss rate from 0.1118% to 0.2862% with one additional path. However when five paths are used between the same source and destination the loss rate falls to 0.1095%.

To characterise the system we adopted a two-state Markov model, as illustrated in Figure

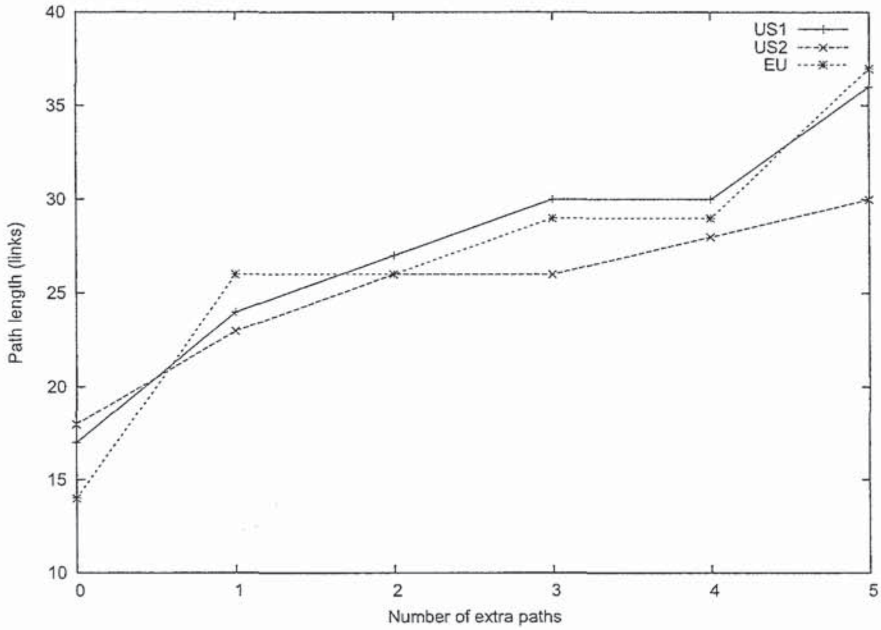


Figure 4.14: Number of links on each path

4.19, where packets are not received when in the Bad state and packets are successfully received when in the Good state. Using the data we gathered from the network we calculated the P and Q values such that P is the probability that the next packet will be lost if the current one has been received, and Q is the probability that if a packet is lost then the next packet will be successfully received.

Figures 4.20 and 4.21 show how the P and Q values would change when path diversity is used. In general, when five paths are used the Q value is improved compared to the single path case, resulting in bursts of shorter duration. In the case of the two US topologies the improvement in Q comes at the cost of P rising too, which will result in more bursts. The change in P and Q values leads to the question of how many path could be in error at the same time. The reasons for multiple paths being in error at the same time could either be because independent links on all paths simultaneously experience error at the same time or because common links are in error.

The initial set up process was based on minimising the number of links between the client and server. The results shown in Figures 4.22 and 4.23 show that in some cases errors are experienced on multiple paths at the same time, while in other cases errors just occur on one path. As an example, in the case of the US1 group, when three extra paths are selected, 98%

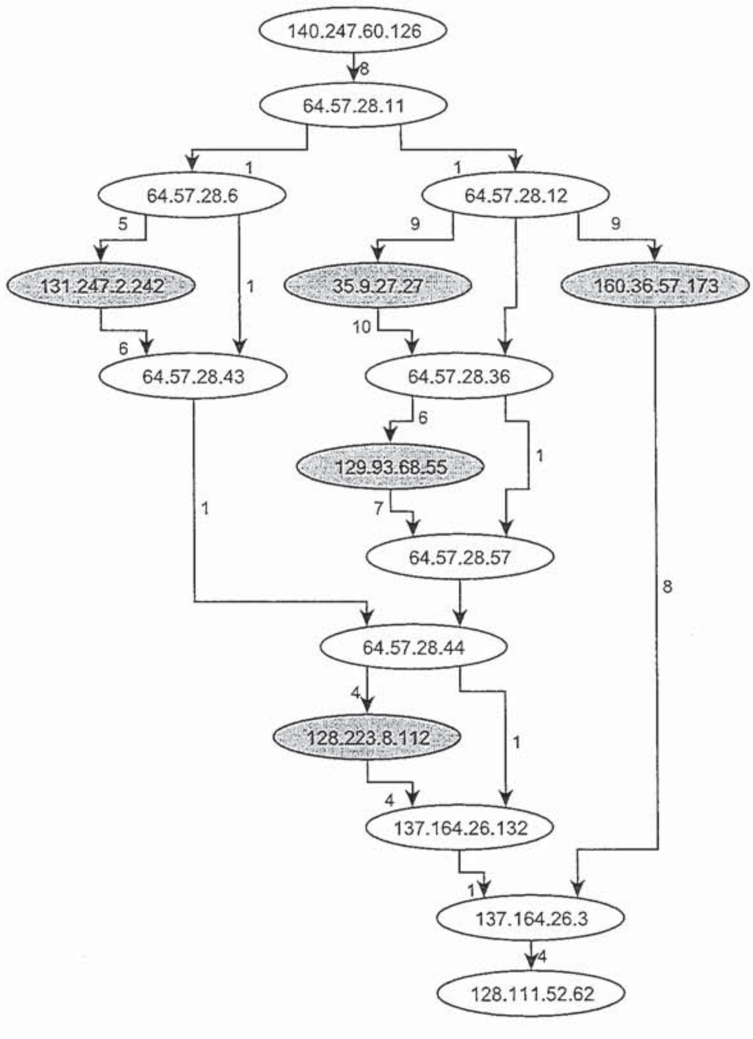


Figure 4.15: The number of links between selected routers and relay nodes (shaded) used in the US1 overlay network

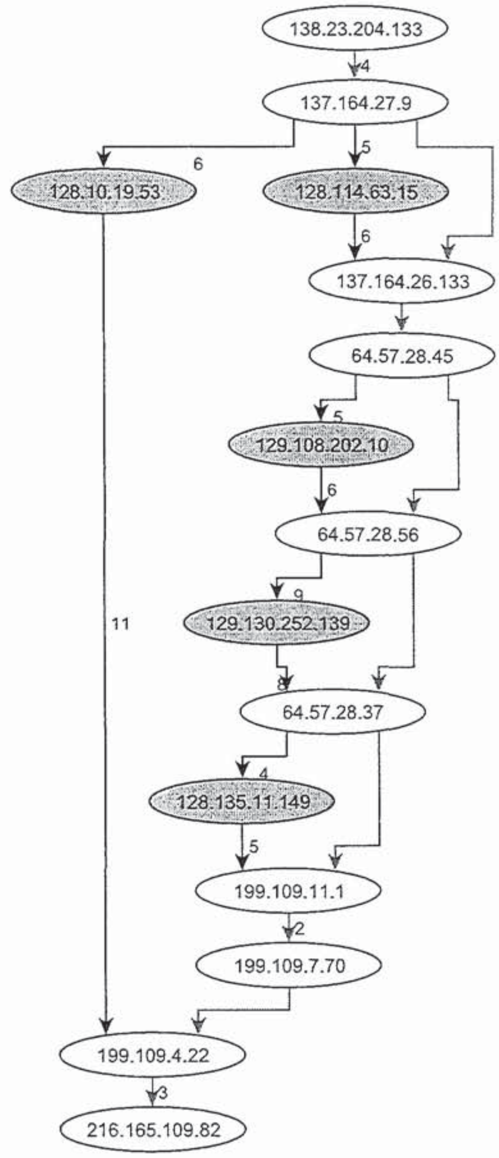


Figure 4.16: The number of links between selected routers and relay nodes (shaded) used in the US2 overlay network

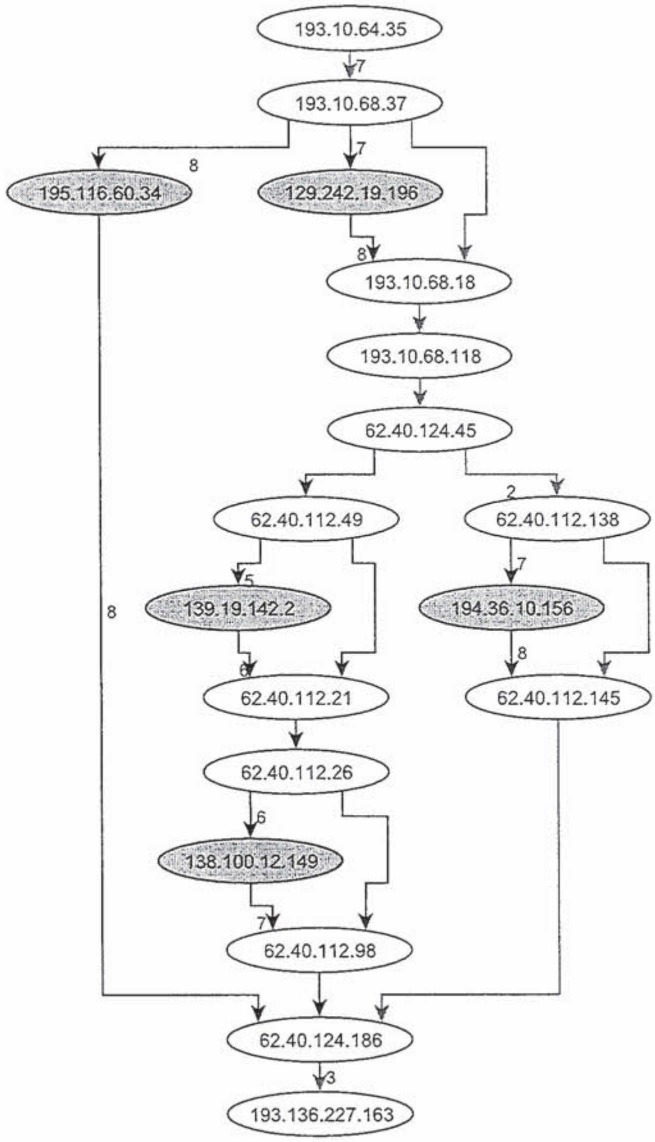


Figure 4.17: The number of links between selected routers and relay nodes (shaded) used in the EU overlay network

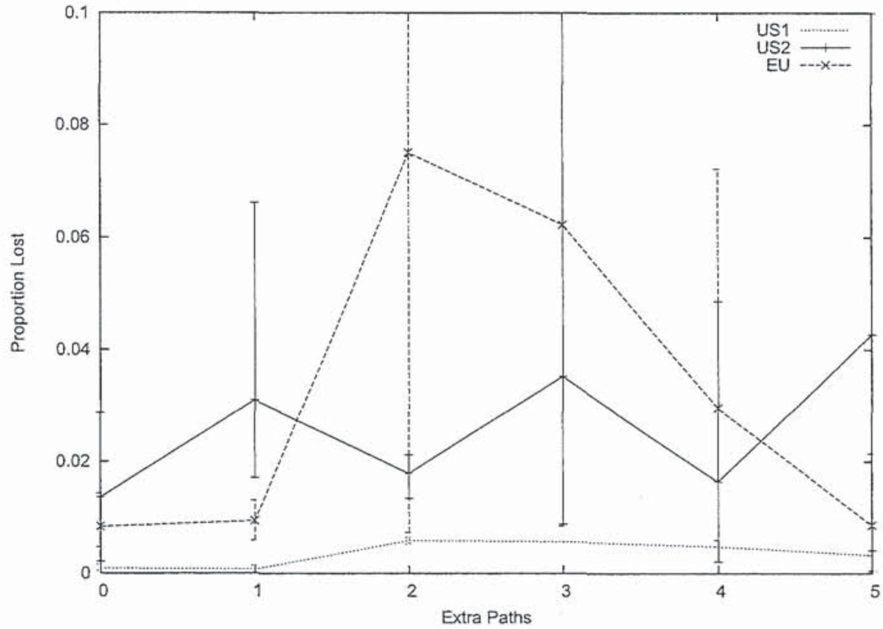


Figure 4.18: Percentage of Packets Lost

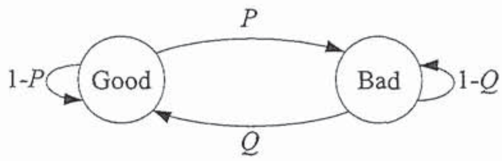


Figure 4.19: Two state Markov chain, burst characterisation for single link

of all losses occur across all four paths at the same time. This is very different from the EU group, where when three paths are used just 1.6% of all losses occur on all paths at the same time.

The change in values for P and Q will affect the number of consecutive losses and the number of consecutive successes. Figures 4.24 and 4.25 show the length of losses in the system. In the case of Figure 4.24, path diversity provides the benefits expected as it reduces the number of packets in a burst loss. However, this is not always the case as it can be seen in Figure 4.25 where path diversity initially increases the number of consecutively lost packets. If the change in the loss percentage for the system is not significant and the length of burst losses are getting shorter, then the length of consecutively received packets must be getting shorter when the burst length is getting shorter.

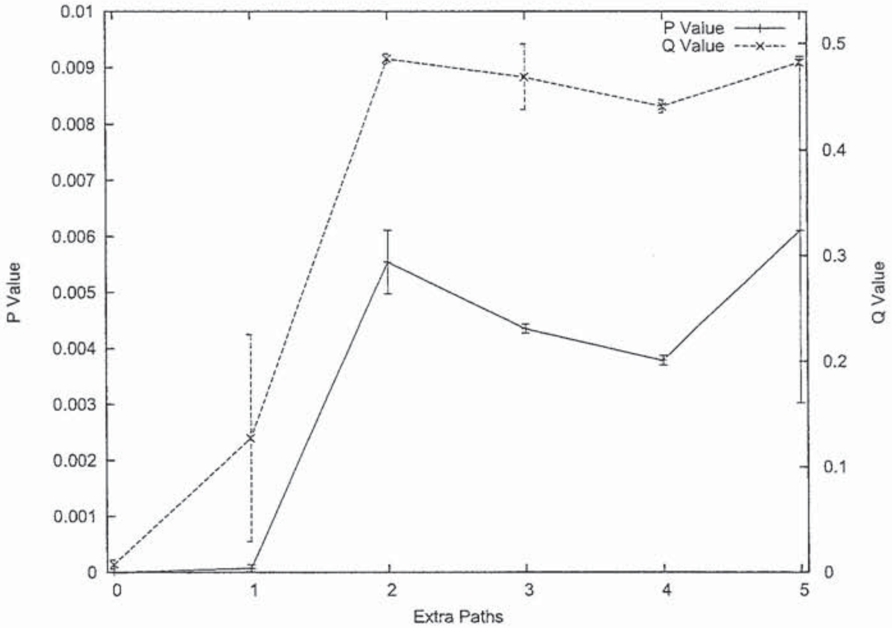


Figure 4.20: US1:P and Q value

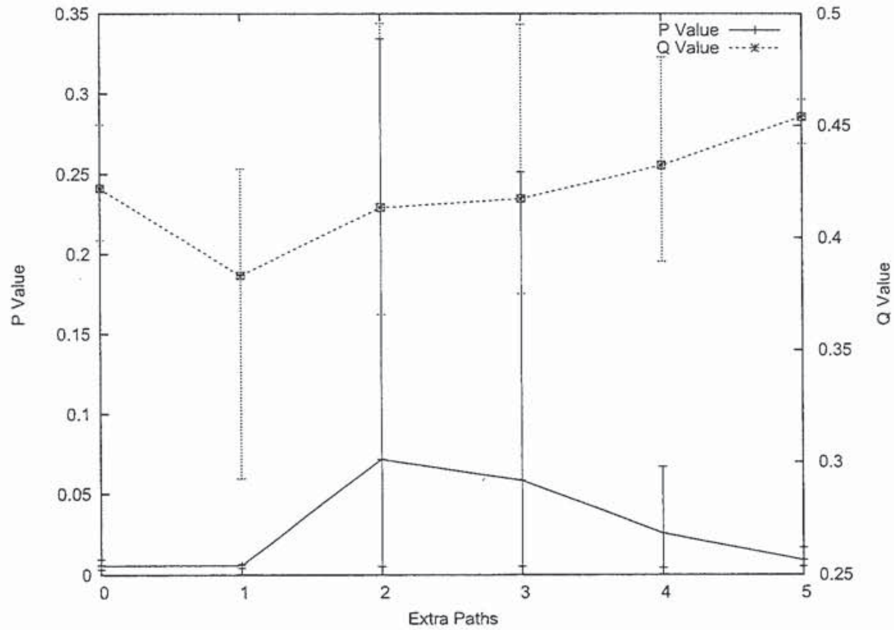


Figure 4.21: EU:P and Q value

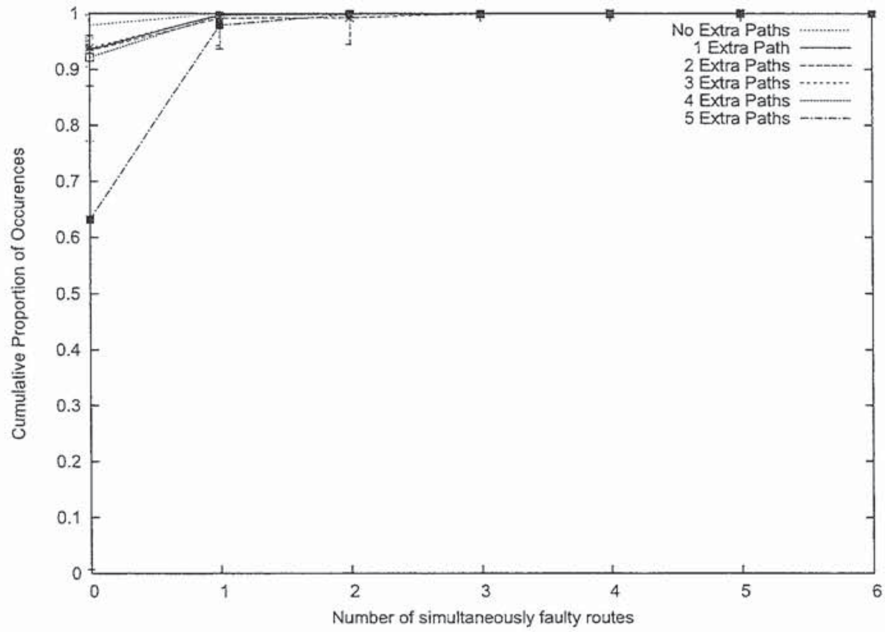


Figure 4.22: US2: Number of simultaneously unavailable paths

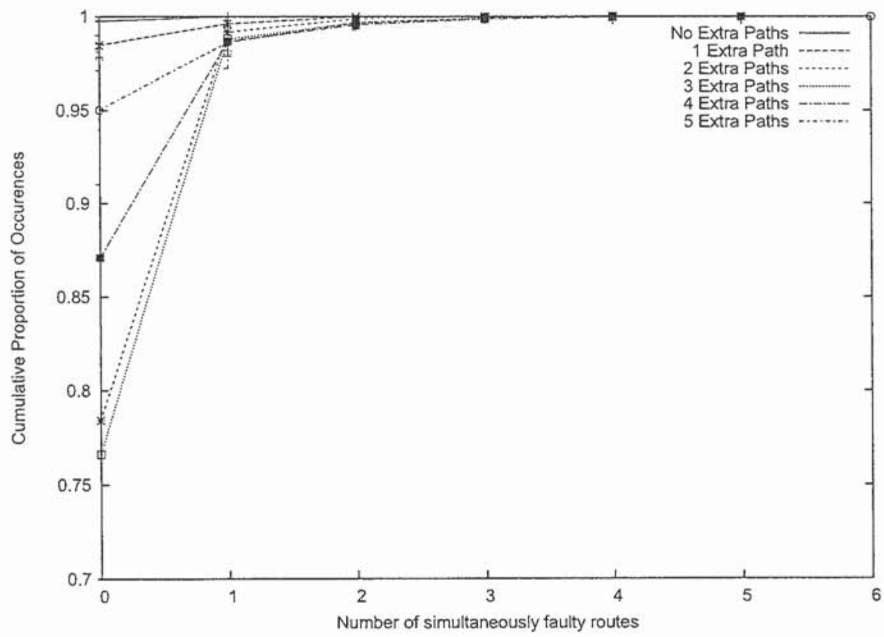


Figure 4.23: EU: Number of simultaneously unavailable paths

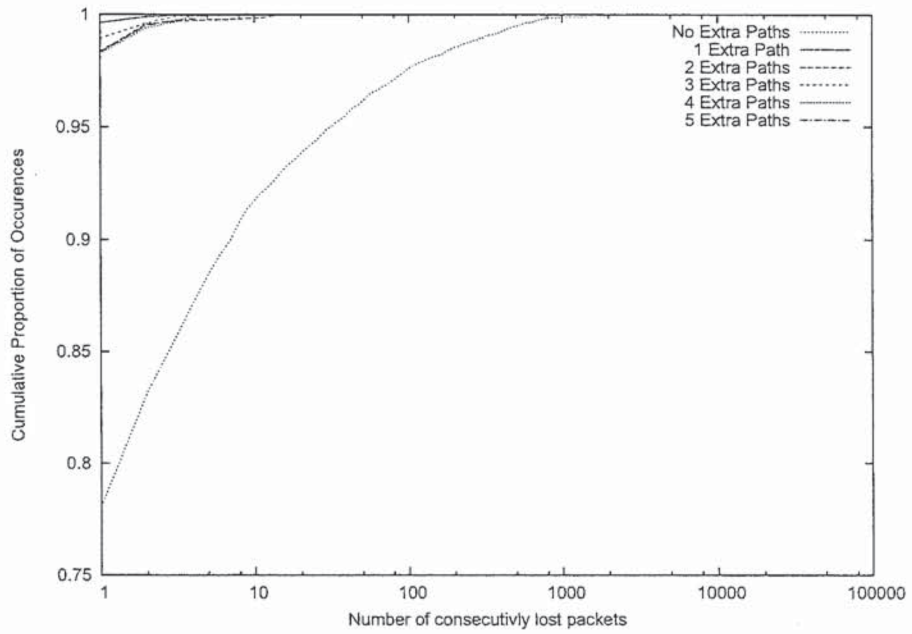


Figure 4.24: US2:Number of consecutively lost packets

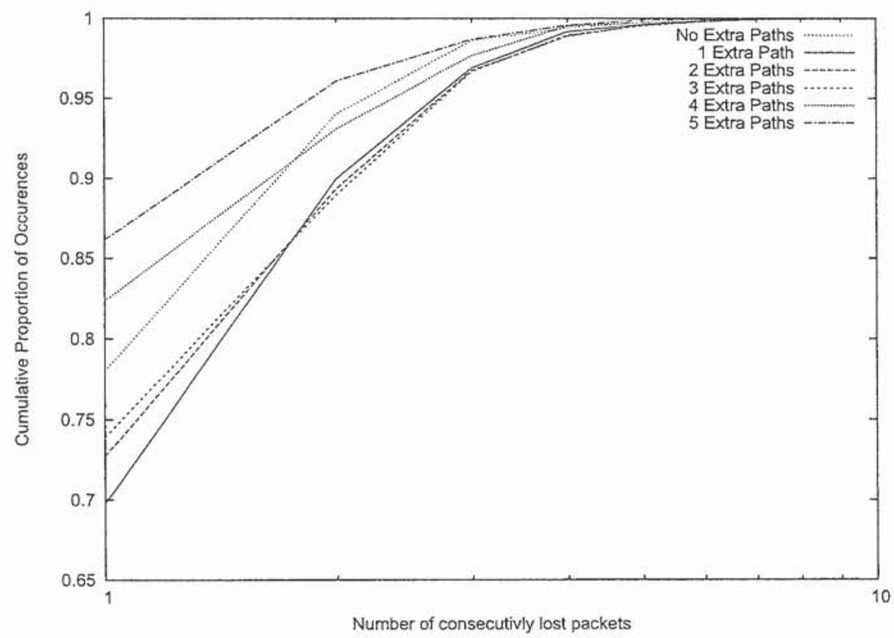


Figure 4.25: EU:Number of consecutively lost packets

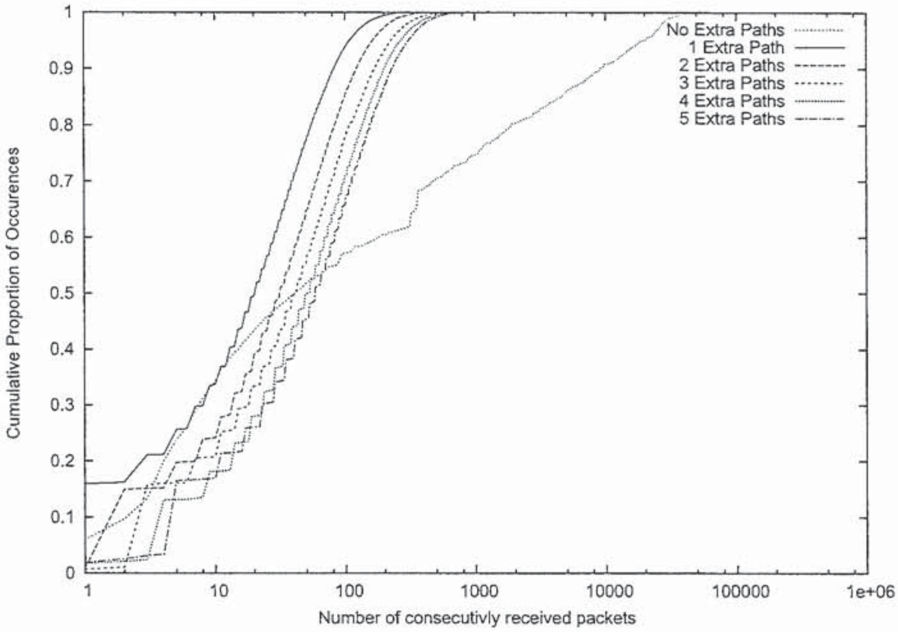


Figure 4.26: US2:Number of consecutively received packets

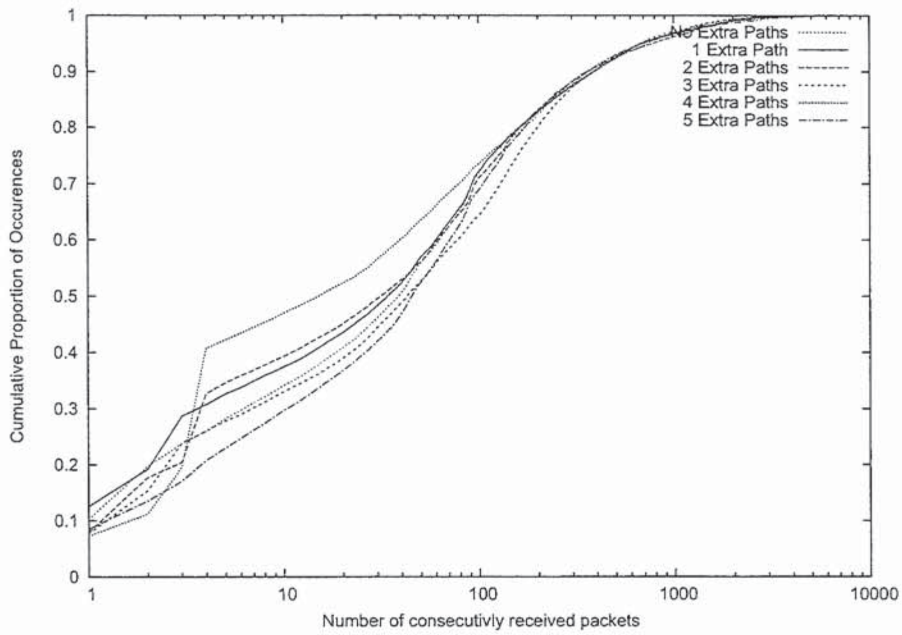


Figure 4.27: EU:Number of consecutively received packets

Figures 4.26 and 4.27 show the number of consecutively received packets. From the results presented in Figure 4.27, we can see that in the single path case the results are approximately the average, with some instances of path diversity increasing the success duration as well as reducing the loss duration. However, in Figure 4.26 there is an initial large drop in the number of consecutively received packets when one path is added but not much change thereafter.

To aid our analysis we calculated the correlation in terms of the ratio of the paths which are correlated (C_N) with other paths in our overlay network, defined by the formula:

$$C_N = 1 - \frac{\sum_{n=1}^{N} \frac{I_{n,N}}{L_n}}{N} \tag{4.9}$$

Where N is the number of paths, L_n is the number of links on path n and $I_{n,N}$ are the number of independent links on path n in the environment with N paths.

Had the network we examined been a homogeneous network then we would have expected to see clearly a correlation between the path independence and the loss probability. However the Internet is a heterogeneous network and as shown in Figure 4.28 there is little correlation. Similarly in Figure 4.29, where the losses occur on all paths of the overlay network at the same time, there is no evidence shown of correlation.

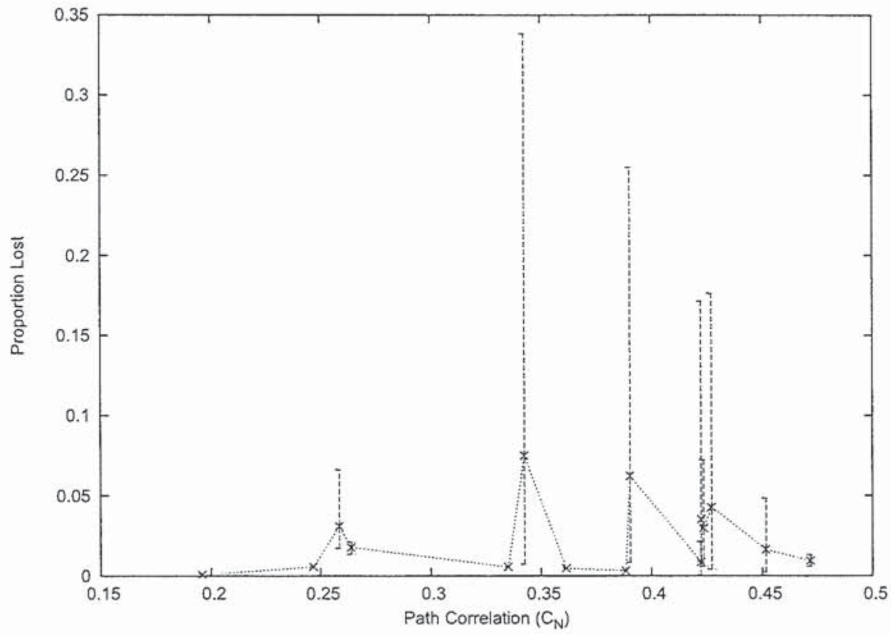


Figure 4.28: Percentage of loss against the path correlation

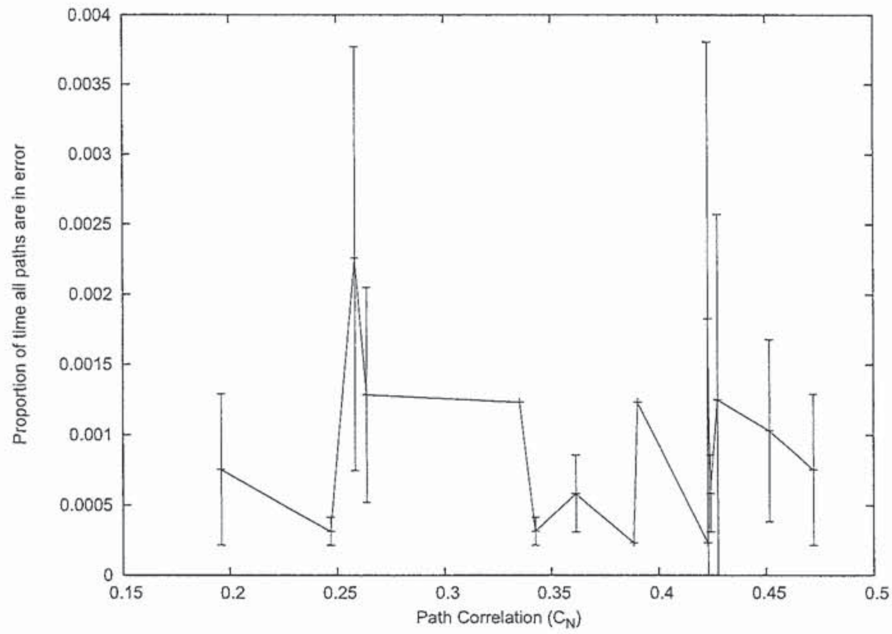


Figure 4.29: The proportion of occurrences of all paths being in error against the path correlation

4.5 Conclusion

In this chapter we have analysed the packet loss probabilities using different loss models (separate and combined) in a path diversity environment. The effectiveness of these models has been confirmed by the simulation results achieved. Most importantly, our results suggest that it is possible to find an optimal number of paths in order to minimise the packet loss rate for the given network conditions.

We have also investigated the packet loss property for the wired-wireless case in four different scenarios. If the causes of loss are the same for both sections then the performance is the same as in the purely wired network. However if the loss patterns are different then the performance could be improved or degraded depending on the primary source of loss.

To identify if our initial simulation results are realistic we have investigated the effect of path diversity on the packet loss properties when UDP packets are streamed across an overlay network on top of the Internet. We assume that the feedback channel is unavailable in our investigation, and that packets are dispatched towards the destination in a round robin manner. The results presented cover the scenarios where up to six paths are selected, based on the selection criterion that the N paths with the fewest number of links are selected among all the paths available in the overlay network.

From the results obtained, we can see that increasing the number of paths could increase as well as decrease the packet loss probability. There are many factors that contribute to the loss properties concerned. In general, if congestion is the main source of packet losses, compared to the non-congestion related (such as random and burst) losses, path diversity can help divert traffic and reduce the loss probability as a result. However, if non-congestion related losses are dominant, path diversity may lead to more losses (this would mean that the benefits of path diversity in a congested environment are being negated due to the increase in non-congested losses).

Chapter 5

Video Streaming of H.264 Data Partitions

In this chapter we initially apply H.264 Data Partitioned video streaming to the path diversity environment. To achieve this we have to assign data partitioned video packets to the available paths. We present a method of dynamically assigning the packets based only on the knowledge which can be gathered from the NAL header (introduced in Subsection 2.2.2.1). The results are compared against a static assignment. The assignment scheme is then extended to include FEC redundancy. We show that with both FEC and path diversity that this can produce benefits in some scenarios.

An increasing number of clients are connecting to the Internet with a wireless last link. Researchers have previously proposed an assignment scheme to improve the quality of video received in such an environment, these were completed through simulation. We set up a testbed using currently available hardware and software. We compare the assignment scheme with single class assignments and show that the latter provides an improved video quality but with a reduction in the overall throughput of the network.

5.1 Video Streaming in an Overlay Network

As previously discussed, in Subsection 2.1.5.1, H.264 data partitioning can separate a video into multiple streams with unequal importance. In this section we propose a mapping scheme which assigns H.264 data partitioned video to the available paths in an overlay network.

5.1.1 Assignment Scheme

In the assignment scheme we first arrange the order of the packets (containing a single partition) based on their importance. Firstly the frames are sorted by their type, i.e. the Key frame first (as a reference for all the frames up until the next Key frame), followed by the Inter coded frame and then the Bi-predicted frame. Secondly the partitions are sorted such that partition A is first, then partition B and finally partition C. Finally the partitions are sorted by the frame sequence number in an ascending order.

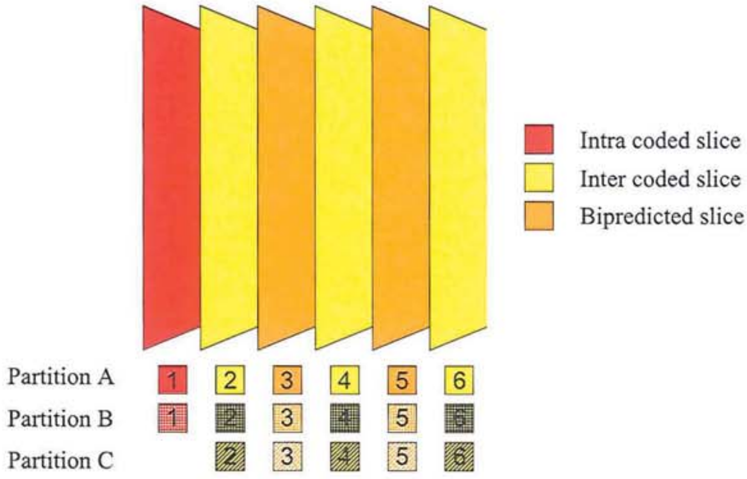


Figure 5.1: Sample H.264 data partitioning with frame numbers

Figure 5.2: Sample H.264 priority partition ordering

For example, using the sample GOP shown in Figure 5.1, we show the sorted importance order in Figure 5.2. Of the sample ordered packets the first two packets are the only packets associated with the Key frame, so with the highest importance and are placed at the top of the order (leftmost); and the partition A packet is in front of the one for partition B. The second group in the order are the packets related to the Inter coded frame and for partition A. There are three packets belonging to this group, and they are placed according to the sequence number of the frame they represent, e.g. 2,4 and 6. The third and fourth groups with four packets each are also related to the Inter coded frame but for partition B and partition C, respectively. In the same way, the remaining groups are related to the Bi-predicted frame and placed according to the ordering process described above.

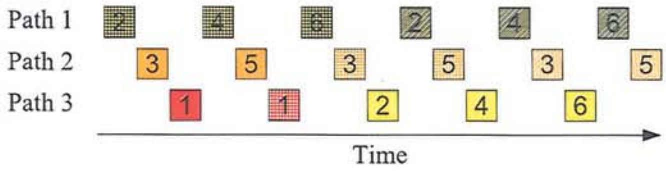


Figure 5.3: Sample H.264 partition assignment

The last step of the scheme is the path assignment that maps the packets in the ordered sequence onto the available paths so that the importance of the packet is matched to the quality of the path. The ordered sequence of packets shown in Figure 5.2 is divided into N groups, where N is the number of paths available. In this example, $N = 3$. We then assign the first group of 8 packets to the best path (path 3 shown in Figure 5.3), the second group of 9 packets to path 1 (middle), and finally the third group of 9 packets to path 2 (worst).

Given the assignment scheme explained previously we term this the static scheme, we also present a dynamic assignment scheme based on the monitoring of the quality of paths and switching the path assignment adaptively to the changes on network conditions.

5.1.2 Testbed Setup

Using the log files previously captured through the PlanetLab experiments we replay the losses onto the video stream. The way this is achieved is shown in Figure 5.4. Within the loss replay system we use JM v13.2 [121] to encode a 10 minute CIF video into a data partitioned H.264 video stream of RTP packets with a GOP size of 72 frames, where each frame was compressed into a single slice with frames within the GOP encoded into alternating B and I slices. The network conditions are replayed using the log files already captured. The video packets are passed through the network emulator which imposes losses within the stream at the positions they would have occurred had the packet actually been streamed across the network. The resulting RTP packets are decoded by JM and then any losses are concealed using motion vector copy concealment. The original, encoded and decoded videos are then processed to compute the PSNR of the video transmitted and the video after concealment.

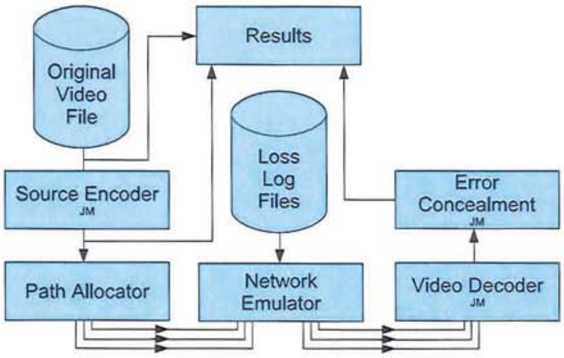


Figure 5.4: H.264 path assignment testbed

5.1.3 Results and Analysis

Since the loss conditions of network paths will change over time the assignment scheme can be applied dynamically, where the last step of assigning packets to paths can be repeated when network conditions change. Since the packets are dispatched in a round robin manner there is no need to send the individual loss rate for each path, instead just an ordered list containing the path with the lowest lost rate path first needs to be sent to the server. Feedback can be minimised by informing the source only of a change in the order path list. The channel conditions are averaged over a number of GOPs, with the conditions fed back to the source before the transmission of the next GOP. The channel sampling is offset by a minimum of half of the RTT, as illustrated in Figure 5.5.

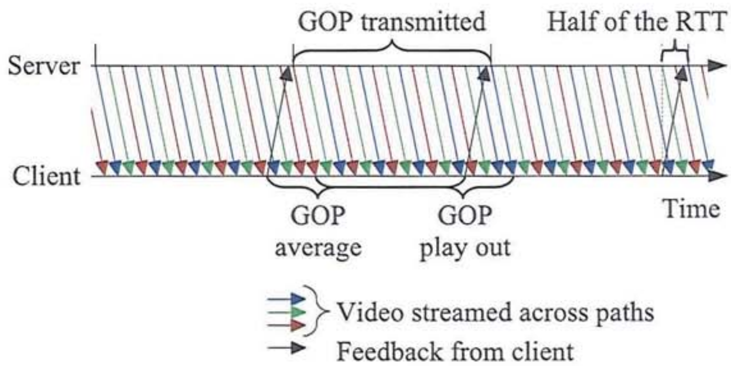


Figure 5.5: Timing relationship of feedback, play out and transmission of packets

Figure 5.6 shows, for the overlay network scenario, the PSNR performance of the dynamic assignment; static assignment and the loss free case are presented. The dynamic scheme shown in Figure 5.6 shows the effect of averaging over different numbers of GOPs. From the scenario shown in the figure, there is a significant improvement in the dynamic scheme over that of the static scheme. In addition, the performance of the loss-free video (subject to compression only) indicates the upper bound of the performance that can be achieved. When errors occur within the video stream the PSNR value will be reduced, this will be indicated in the graphs by a shift towards the left.

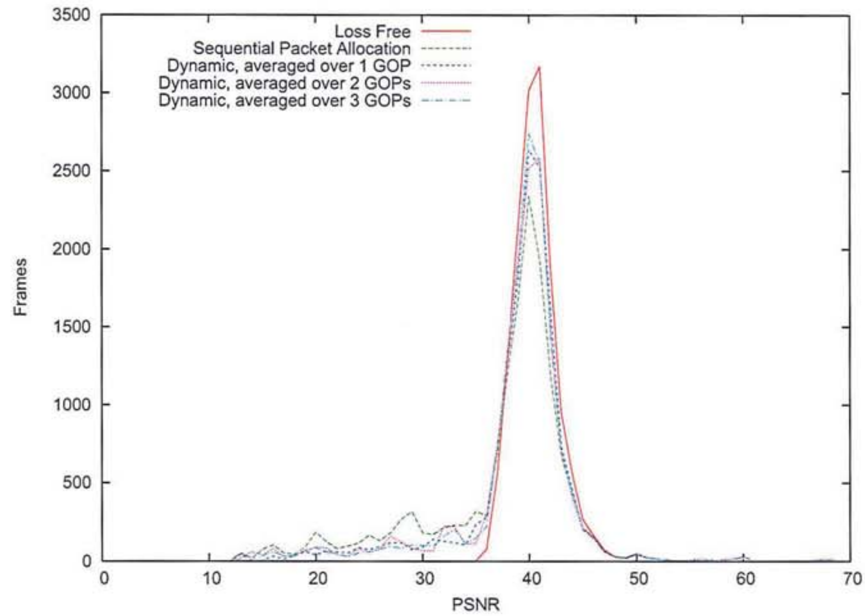


Figure 5.6: A sample set of results for H.264 assignment when five paths are used

Figures 5.7 and 5.8 show, for the EU and US2 topology respectively, the average number of frames of each PSNR value, where the PSNR value of each frame is rounded down. Figure 5.7 shows that the use of path diversity can improve the quality of the stream received by the client, by increasing the number of frames with higher PSNR value in all scenarios. This is in contrast to Figure 5.8 where path diversity is unable to produce any benefits.

These features highlight the complexities experienced when implementing an overlay network. It is not always possible to improve the quality of the video stream received using path diversity. This is because using past events to predict future ones is not always accurate.

Feedback is only sent from the client to the server to indicate when the network conditions have changed. When the network conditions remain constant no updates are sent.

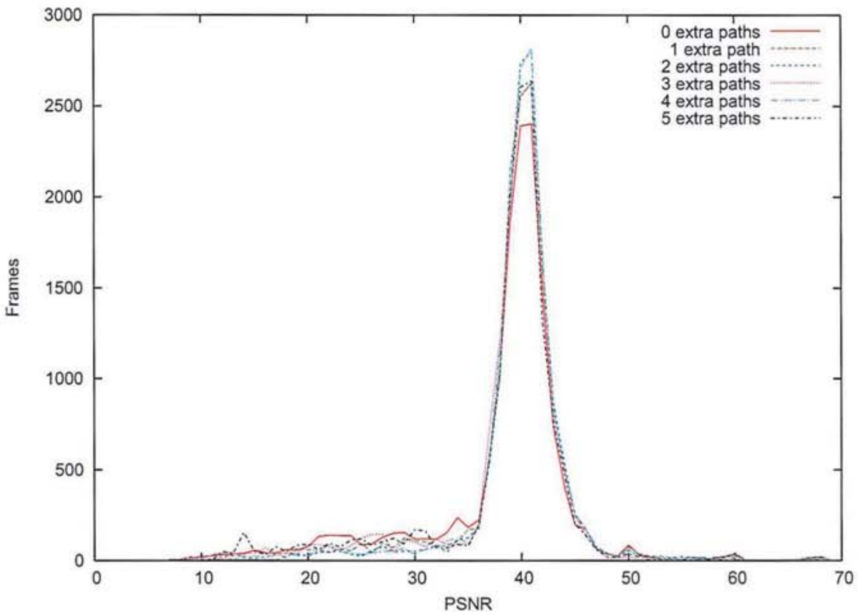


Figure 5.7: EU:The average effect of streaming a 10 minute video with dynamic path allocation averaged over one GOP

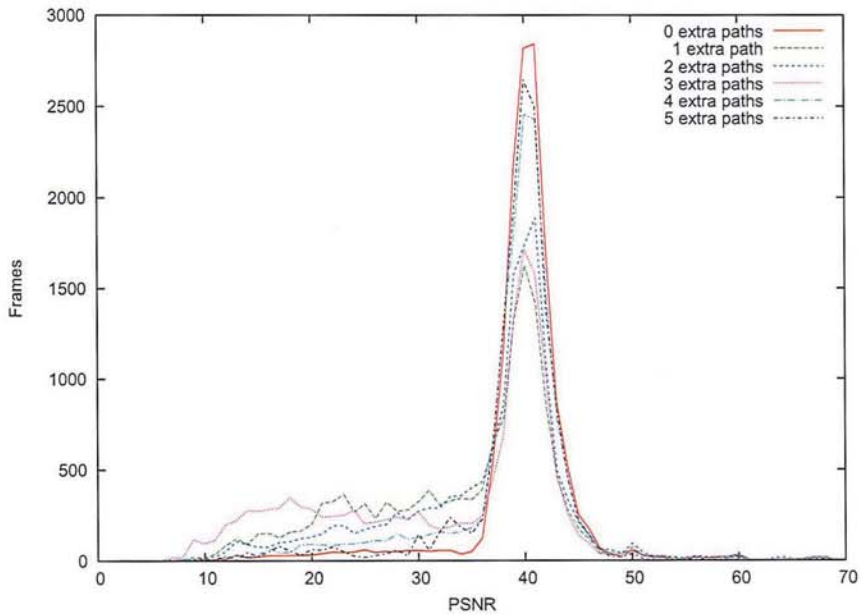


Figure 5.8: US2:The average effect of streaming a 10 minute video with dynamic path allocation averaged over one GOP

Therefore the stability of the network paths, i.e. the amount of time the order remains the same, can be inferred from the feedback overhead which is defined as the ratio of packets sent from the client to the source, against those sent from the source to the client. Figure 5.9 shows that the feedback overhead for topology US2 can be twice that of the EU topology. From this we know that the EU topology has fewer path re-orderings than in the US2 case. It is the re-orderings that reduces the effectiveness of the path diversity scheme.

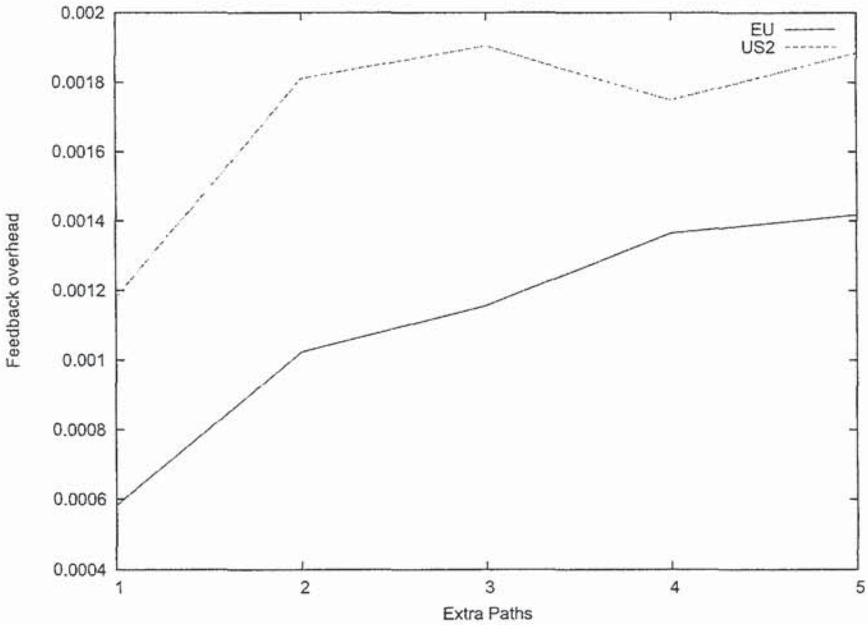


Figure 5.9: The dynamic feedback overhead

5.1.4 Unequal Forward Error Correction

At the packet level, FEC codes can be used to correct erasures [122]. The difference between correcting errors and correcting erasures is that in an erasure environment the location of the error is known. In a network each packet contains a sequence number, so it is possible to identify which packets are lost and which are received.

With the use of the maximum distance separable (MDS) block based FEC codes, such as Reed-Solomon (RS) [123], k original packets can be encoded into n encoded packets. When any k encoded packets are received at the destination the original k packets can be reconstructed. It is thus possible to recover the original data when up to $\frac{n-k}{n} \times 100\%$ of packets are lost.

When using FEC there are two different encoding options, systematic and non-systematic. In the systematic case the redundancy is added to the existing data. In the non-systematic case the original data is encoded into n completely new packets, as illustrated in Figure 5.10. When either a systematic or non-systematic MDS code experiences no more than $n-k$ losses all of the original data can be completely recovered. When more than $n-k$ losses occur, the non-systematic code is unable to recover any of the original packets, whereas the systematic code is able to return the received original packets to the end user. This feature is highlighted in Figure 5.10.

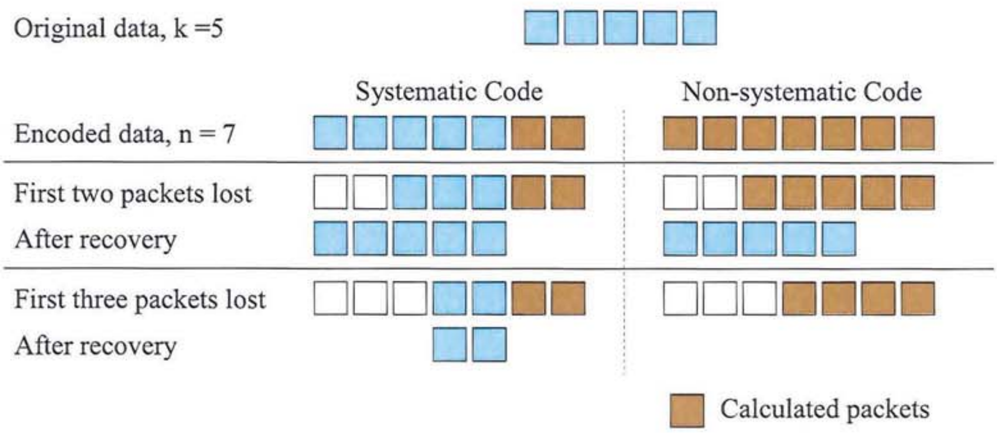


Figure 5.10: Illustration of systematic and non-systematic FEC

For data transmission, the selection of a systematic or non-systematic code is not important since other mechanisms, such as automatic repeat request (ARQ), need to be used to ensure the arrival of all the data. In a video streaming scenario the arrival of some of the packets is better than none. Due to time constraints ARQ can not be used. If a non-systematic code were used for video streaming then there could be large portions of the video unavailable at the receiver at play out time.

Since data partitioning creates streams with different importances we can assign unequal amounts of redundancy to each of these streams. The use of unequal protection increases the likelihood that the most important packets, which are better protected than others, will be available at the client. If we assume that packets are dropped randomly then the probability that we lose $n-k$ packets or fewer out of n, i.e. the probability of receiving k or more out of

n, is equal to

$$\sum_{i=0}^{n-k} \binom{n}{i} p^i (1-p)^{n-i} \tag{5.1}$$

where p is the random packet loss probability, with k and n as the parameters for the MDC code used. Figure 5.11 shows the probability that k packets or more are received, where $k = n-1$ and $k > 0$ thus n must be > 1.

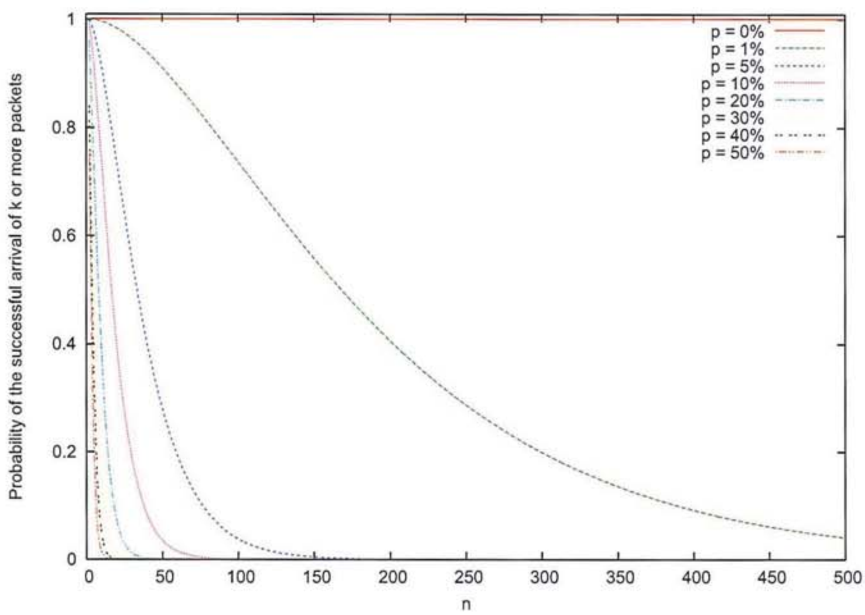


Figure 5.11: Probability of receiving k or more of the n packets transmitted, where $k = n-1$, with assorted random packet loss probabilities, p

From Figure 5.11 we see that a smaller value of n increases the likelihood that k or more packets arrive. Traditionally, large values of n are used in order to keep the additional overhead, also termed the redundancy rate, down. The redundancy rate, which is shown in Figure 5.12, is calculated from

$$\eta = \frac{n-k}{n} \tag{5.2}$$

In the multi path environment we have seen alterations in the duration of successes and losses, which are very important when FEC is used. In Figures 5.13 and 5.14 we show the results if a systematic FEC code is used with a block size of $n = 30$ when streaming data

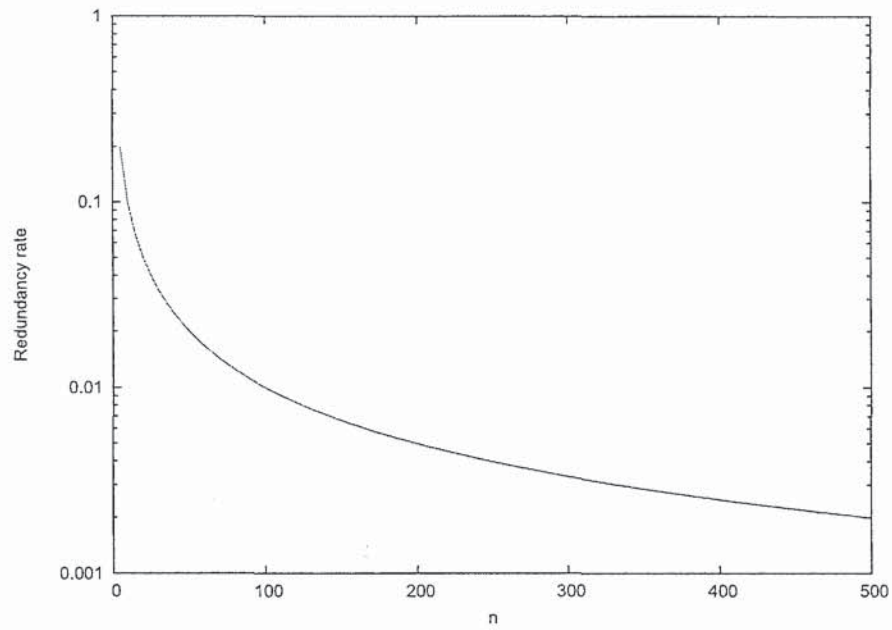


Figure 5.12: Redundancy rate from the use of FEC where $k = n - 1$

over our planetlab network. The results presented in these two graphs show that FEC can work better when used in conjunction with path diversity than when FEC is used on just a single path.

Due to the unequally important nature of video partitions, protecting the most important part of the video means that the overhead remains constant while the parameter n is reduced for the most important partitions. With the introduction of FEC we adjust the previous assignment scheme. We create priority class, 1 to 5, in priority level 1 we classify all the packets relating to the key frame and in priority class 5 we classify all the packets relating to the bi-predicted frames. The reason for grouping all the partitions from the key frame is because there are not many packets but they are the most important, if redundancy were added to partitions A and B individually then the additional overhead for these packets would be 100% by grouping the partitions reduces the overhead to 50% without a significant change in performance. The bi-predicted frames are the least important and we group these together, this is again to provide more efficient use of redundancy. In classes 2, 3 and 4 we assign partitions A, B and C from the inter coded frames respectively. FEC packets are added per priority level, as shown in Figure 5.15. The number of redundant packets added to a priority class is independent of the redundancy added to other priority classes. We use the notation

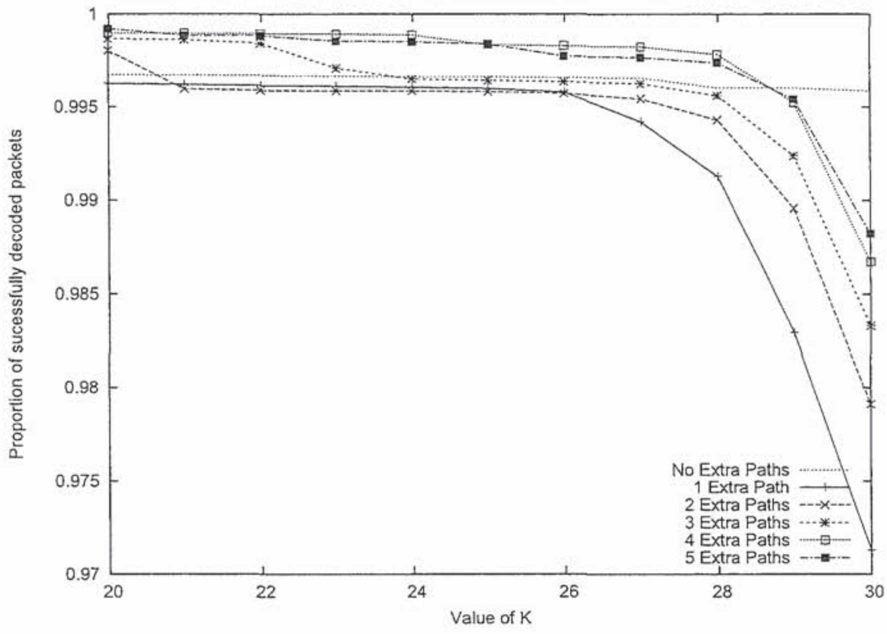


Figure 5.13: US2:The effect of changing the value of k on the proportion of packets received

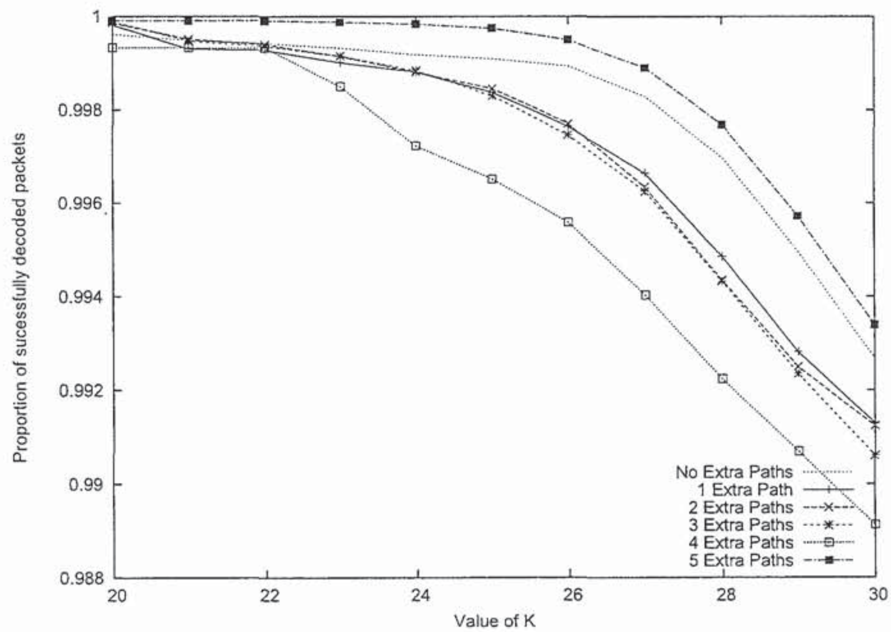


Figure 5.14: EU:The effect of changing the value of k on the proportion of packets received

$\{R_1, R_2, R_3, R_4, R_5\}$, where R_l is the number of redundancy packets added per GOP, assigned to priority class l where $1 \leq l \leq 5$.

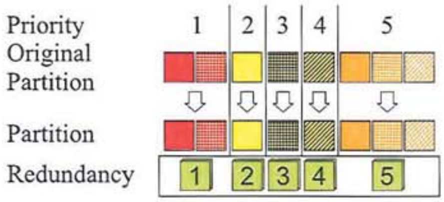


Figure 5.15: H.264 priority class assignment with unequal FEC redundancy

All redundancy is assigned to a new priority class 6. The generation of an ordered stream of partitions, including redundancy, is created in the same manner as when no FEC was included. With the same path ordering as before (path three had the lowest loss and path 2 the highest) the ordered packet stream is distributed as shown in Figure 5.16.

Figure 5.16: Sample H.264 priority class ordering including FEC redundancy

The ordered packet sequence is then assigned to the paths in the same manner as without FEC, a sample assignment is shown in Figure 5.17.

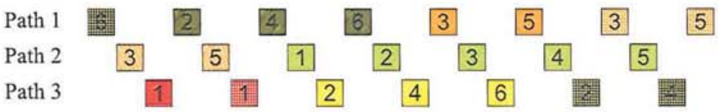


Figure 5.17: Sample H.264 priority class ordering including FEC redundancy

To incorporate the FEC encoding and decoding our testbed is altered. Figure 5.18 shows the revised testbed system.

Figure 5.19 shows the PSNR distribution when both priority classes 1 and 2 both have an additional redundant packet, i.e. an FEC scheme of $\{R_1, R_2, R_3, R_4, R_5\} = \{1, 1, 0, 0, 0\}$. These results show, for the EU topology, that path diversity can improve the effectiveness of the FEC code used.

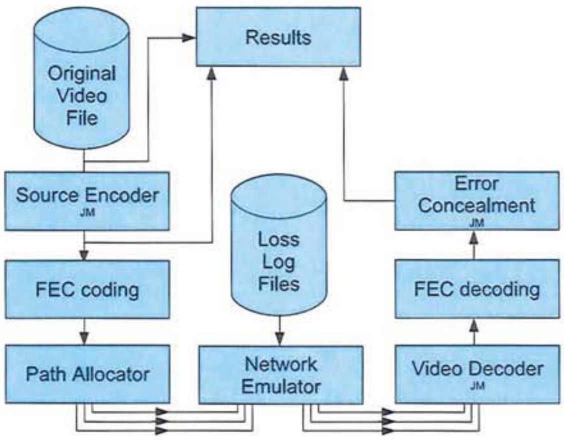


Figure 5.18: H.264 path assignment testbed

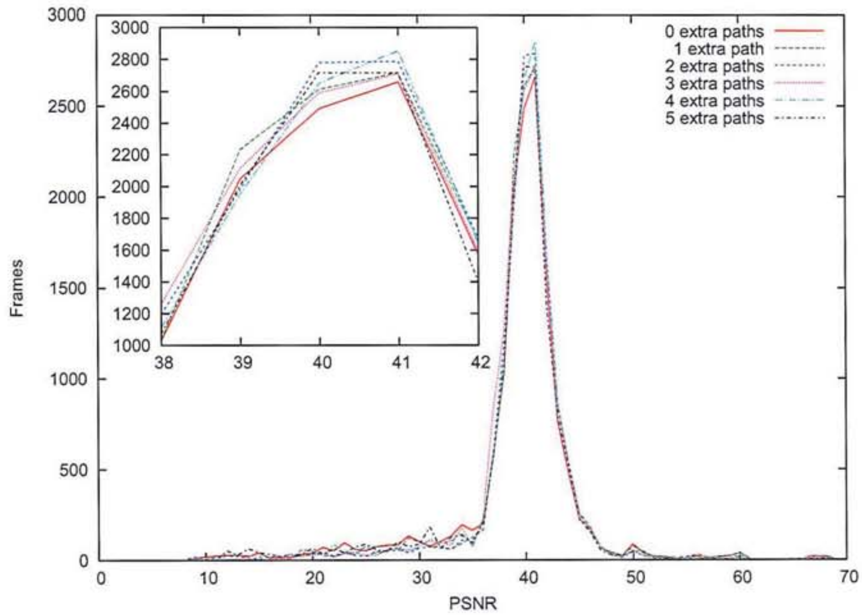


Figure 5.19: EU:Improvement of using path diversity with a $\{1,1,0,0,0\}$ FEC scheme

Figure 5.20 presents the results for the US2 topology, with the same parameters as those in Figure 5.19. It can be seen that in this scenario FEC with path diversity provides no benefits, compared to the case without FEC.

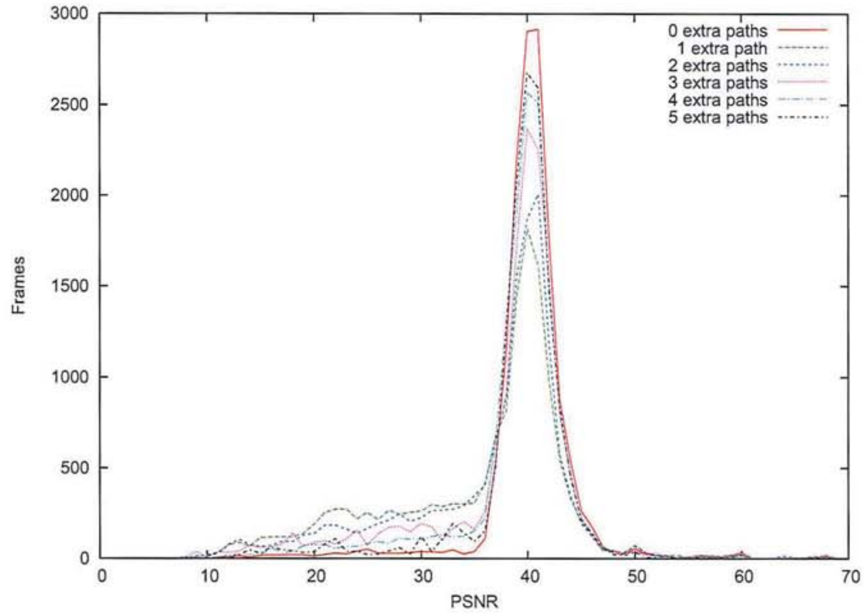


Figure 5.20: US2:Impact of using FEC with a $\{1,1,0,0,0\}$ FEC scheme

We now focus on the case where FEC can provide benefits. In Figure 5.21 we show the effect of using the unequal FEC assignment scheme. The addition of a single packet is equal to an average overhead of 0.53%. The overhead is an average as there are not always identical number of packets in a GOP, even though the number of frames remains constant.

Figure 5.21 shows that as the number of redundant packets increases as the average PSNR value increases, and at the same time the standard deviation reduces. These results tend to the values for the loss free scenario where the PSNR averages 40.22dB with a standard deviation of 2.67dB.

Figure 5.22 shows how the proportion of frames received at the client, which are not identical to those after compression, reduces with the use of FEC. Without FEC around 7% of frames displayed contain errors. When FEC is used with six redundant packets, this is reduced to 0.45%.

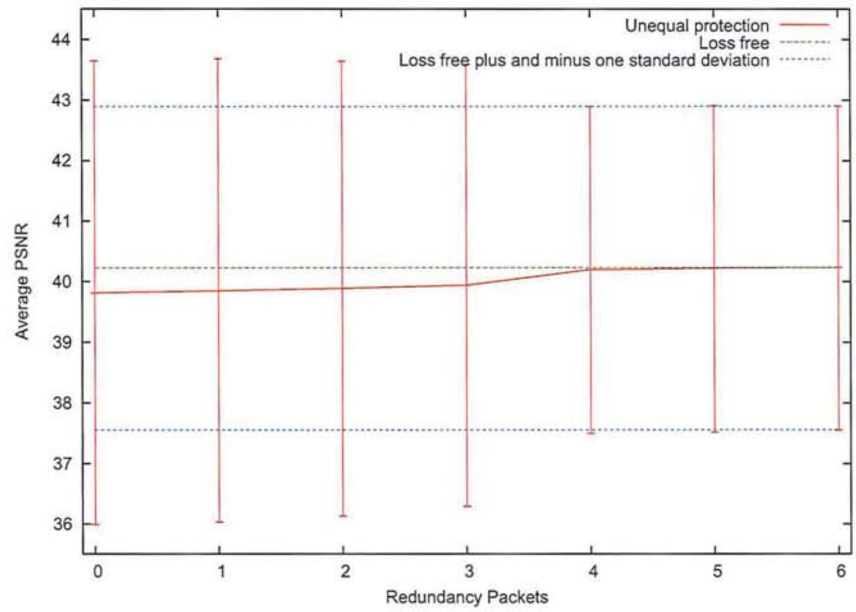


Figure 5.21: EU: Improvement in PSNR with use use of FEC

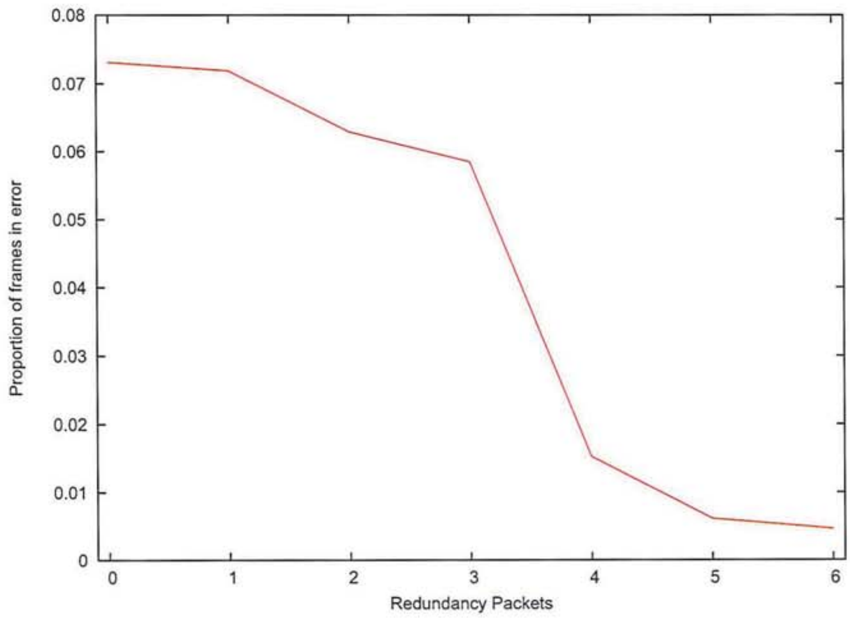


Figure 5.22: EU: Improvement in the proportion of frames displaying errors when using FEC

5.2 Video Streaming in Wireless Local Area Networks

Increasingly, the last link to a client's PC is wireless. In Subsection 4.2.5 we highlight the issue of common links when used with path diversity. To ensure the best quality is received by the client, certain techniques for streaming over the wireless link needs to be considered.

In this section we show, using a real testbed, that different AC assignment schemes will have diverse impacts on the quality of a video stream received by a client in WLANs. We also compare the multi-class assignment scheme proposed by the authors of [124] with the existing single class assignment scheme and find that the latter can produce better results in the tested environment. Additionally, we show how virtual contention at the access point can, in some circumstances, cause a greater number of packet losses than the wireless medium itself.

A number of previous works have investigated the assignment of H.264 data partitions to different wireless network access classes. However, the majority of them were completed through simulation. The authors of [124] presented the QoS arch scheme and simulated a H.264 video stream containing both IDR pictures, i.e. frames compressed into solely intra coded slices, and predicted slices. They show that by allocating partitions to access classes, in accordance with the packet's importance, the loss percentages of the IDR and partition A packets can be reduced. This improvement comes at the cost of an increase in the number of losses in partitions B and C. For comparison we compare the schemes for IDR and inter coded slices.

Some recent experimental work [125] investigates the effect of varying the TxOP parameter on the video quality when using data partitioning. The TxOP parameter allows for a station to send a burst of wireless frames at once in a single contention attempt. This benefits streams, such as video, where the packets are dispatched in bursts. For example, IDR packets are large and can span a number of data frames. These could be transmitted in a single TxOP. In our work the standard TxOP parameters are used (as shown in Table 5.1). The allocation of packets into the ACs is the main area of investigation because changing the default parameter set dynamically would require significant modification to the standard protocol.

Current differentiated QoS mechanisms, such as DiffServ and EDCA, only provide service differentiation on a class basis. A common problem at the edge of the network is called

Access Class	Priority	CW_{min}	CW_{max}	AIFS	TxOP
AC_BK	Lowest	15	1023	7	0
AC_BE	Low	15	1023	3	0
AC_VI	High	7	15	2	3008 μs
AC_VO	Highest	3	7	2	1504 μs

Table 5.1: Access Class Parameters

class hijacking, where all traffic is marked as the highest priority in order for some users to gain unfair advantages over other users. However, the class hijacking approach reduces overall QoS, as there is no differentiation between competing traffic types. Our work aims to highlight the importance of correctly classifying packets transmitted in the wireless network, so that the overall QoS for the network as a whole can be increased.

5.2.1 Combined Strategy

For improving the quality of the video received by a wireless client we implemented a cross-layer approach using a real world testbed, in contrast to the previous work presented in [124] where the system is implemented through simulation. In the testbed an indication of the importance of the packet is provided to lower layers in the stack from the video streaming server. This is achieved through the use of the DSCP field within the IP header. We can then set the DSCP field from our serving application and vary its value depending on the packet payload. Using the knowledge we have about the contents of the packet, which can be extracted from the NAL unit header, we are able to set the DSCP value such that when the packet is passed down to the data link layer of the stack, the DSCP field is then mapped to the access classes previously defined.

In the combined strategy, termed QoS Arch, the control information is assigned to AC_VO. IDR frames and Partition A of the predicted slices are assigned to AC_VI with partitions B and C assigned to AC_BE.

5.2.2 Testbed Setup

Using JM v13.2 [121] we encode a 5 minute video sequence into a sequence of RTP packets. We alter the way JM generates NAL octets so that it is possible to identify the frame type and partition type from the NAL unit octet, as shown in Figure 5.23. The standard only states that an NRI must be set to 00 for frames not used as a source of predictions [28]. Hence the alterations made to the NRI mappings is shown in Table 5.2 which enables the identification of the slice type contained within the packet. The partition type can be identified from the type field, shown in Table 5.3.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+-+-+-+-+-+-+-+-+
```

Figure 5.23: H.264 NAL header

Content of NAL unit	Recommended NRI (binary)	Proposed NRI (binary)
I slice	10	11
P slice data partition A	10	10
P slice data partition B	01	01
P slice data partition C	01	01
B slice	00	00

Table 5.2: H.264 adapted NRI mapping

Content of NAL unit	Type
Data partition A	2
Data partition B	3
Data partition C	4

Table 5.3: H.264 type mapping

A CIF size video is encoded with a GOP of length 36 frames which is initially stored as a file containing the sequence of RTP packets. The number and size of the RTP packets are shown in Figure 5.24. We create a server which takes the pre-coded RTP packets and dispatched them according to the details contained in the NAL octet [1] and RTP header.

We use a medium specification desktop PC running Debian Linux (2.6.19 kernel) equipped with a Fast Ethernet LAN and Atheros 5001X+ wireless card to create an AP. The MadWifi 0.94 drivers are used in the Master mode along with the Linux bridging functions. CPU load is monitored during testing and the AP was not considered to be a bottleneck. The medium specification laptop clients are also Debian Linux based, using the same Atheros 5001X+

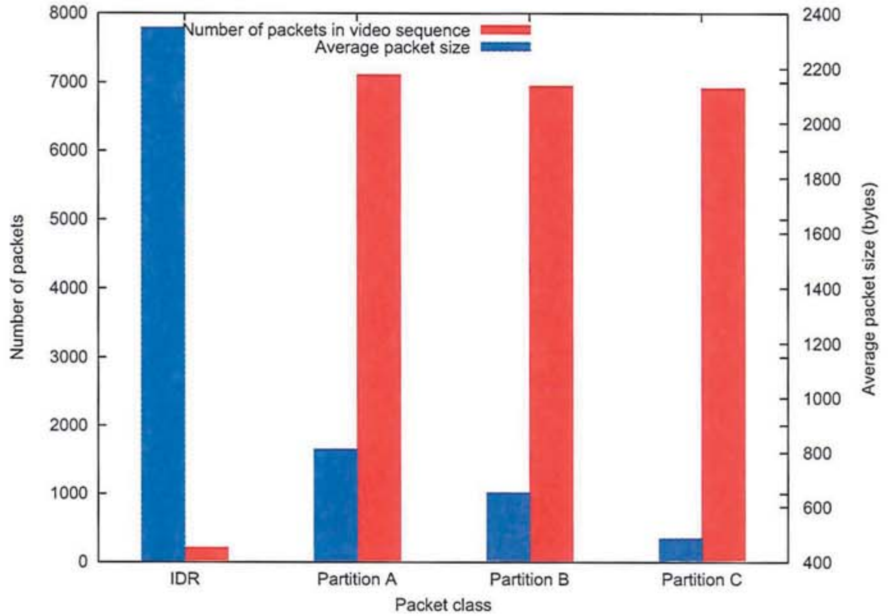


Figure 5.24: Number and average size of the packets for each packet type

cards. The data transmission rate for the AP and clients is fixed at 54Mbps using the 802.11g physical layer, disabling the auto fall back mechanism. Request to send (RTS)/clear to send (CTS) and specific enhancement features such as Turbo G and extended range are disabled. EDCA parameters on the access point are left at default as prescribed in Table 5.1.

Underlying TCP traffic is generated using Iperf 2.02 [126] and placed in the AC_BE priority class, we term this traffic as best effort TCP traffic. For TCP traffic we use TCP Reno combined with a 64k receiver window and a dummy data payload of 1460 bytes. The TCP traffic profile generated by Iperf is indicative of a bandwidth hungry protocol such as FTP or HTTP downloading.

The raw 802.11 packets are captured from the wireless channel using an AirPcap [127] wireless sniffer and tshark protocol analyser running on an additional PC. The raw packet captures are filtered in Wireshark to isolate the UDP/RTP video stream from the background TCP traffic. From this we are able to produce the MAC layer retransmission distribution by analysing the 802.11 frame sequence numbers. The same data is also used to extract the RTP packet numbers for calculating the number and type of the packets sent by the AP.

The testbed network topology is shown in Figure 5.25, where three laptops are placed approximately 5M away from the access point (labelled C1 to C3), without direct line of sight,

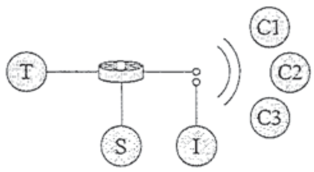


Figure 5.25: Testbed setup

but with no major obstacles, such as walls, in the way. The laptops are used to represent a potential home wireless network, where video and other services may be used simultaneously.

Node T, in Figure 5.25, is the IPerf server, with node S being the video server. Node I is connected to the AirPcap wireless capture card and captures the packets sent from the access point. To ensure that the majority of packets are captured the AirPcap device is attached to the access point.

Using the previously mentioned testbed we complete a series of experiments. Firstly, we stream the video to a single client over a network using the legacy DCF protocol, where all the video and other traffic contend equally for the medium. Secondly, the video is assigned to just a single class using EDCA and streamed over the network. Finally, the packets are assigned across three EDCA access classes according to their importance, as shown in Table 5.4.

	DCF	AC_BK	AC_BE	AC_VI	AC_VO	QoS arch
AC_BK		All				
AC_BE			All			P Partitions B & C
AC_VI				All		I Slice & P Partition A
AC_VO					All	Control
DCF	All					

Table 5.4: Access Class Assignment

5.2.3 Results

We find that when the video is sent in the AC_BE class along with the TCP background traffic that the best throughput is obtained for the best effort TCP traffic, which is shown in Table 5.5. When the video is allocated to the AC_VO class the worst throughput for the best effort TCP traffic was seen.

Assignment	Average Best Effort Throughput
AC_VO	10.23 Mbps
AC_VI	13.19 Mbps
AC_BE	14.10 Mbps
AC_BK	13.44 Mbps
Quad	13.35 Mbps
DCF	12.28 Mbps

Table 5.5: Madwifi Average best effort throughput

	IDR	Partition A	Partition B	Partition C
AC_VO	100.000%	99.972%	99.986%	99.971%
AC_VI	100.000%	99.986%	99.986%	99.986%
AC_BE	99.502%	99.390%	98.905%	99.662%
AC_BK	99.834%	99.948%	99.914%	99.947%
QoS arch	99.834%	99.972%	99.894%	99.947%
DCF	98.673%	99.127%	98.910%	99.102%

Table 5.6: Percentage of packets transmitted by the access point

In all the cases of AC_BK assignment, the video decoder is unable to decode all of the frames within the sequence. The average decodable duration is 1.88 minutes. In all other scenarios the entire video is decoded. The reason for the video decoder to fail is that when adequate information is not received the decoder is unable to operate.

Figure 5.26 shows that by replacing DCF with any access class other than AC_BK an improvement in the PSNR value of the video can be achieved. Other than AC_BK, assigning the packets to the background class produces the biggest drop in picture quality. The reason for this is the larger AIFS and contention window being used in the AC_BK class. Table 5.6 highlights the issue of virtual contention at the access point, which displays the percentage of packets actually transmitted by the access point. It can be seen clearly that in the cases where all the packets are sent in the same access class (i.e. AC_BE and DCF) the percentage of video packets dropped by the access point is higher than in the other assignments cases.

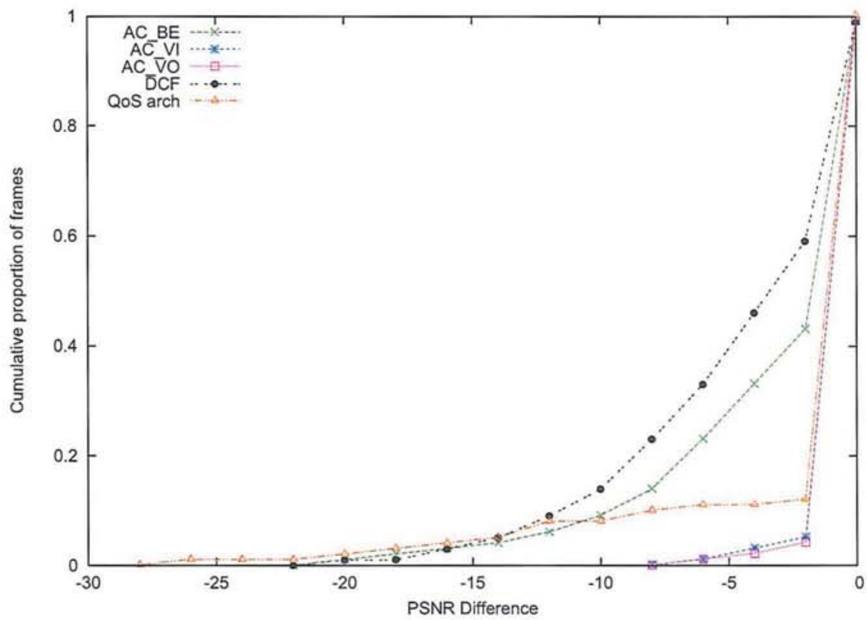


Figure 5.26: The difference in the PSNR of the received video compared to a loss free video

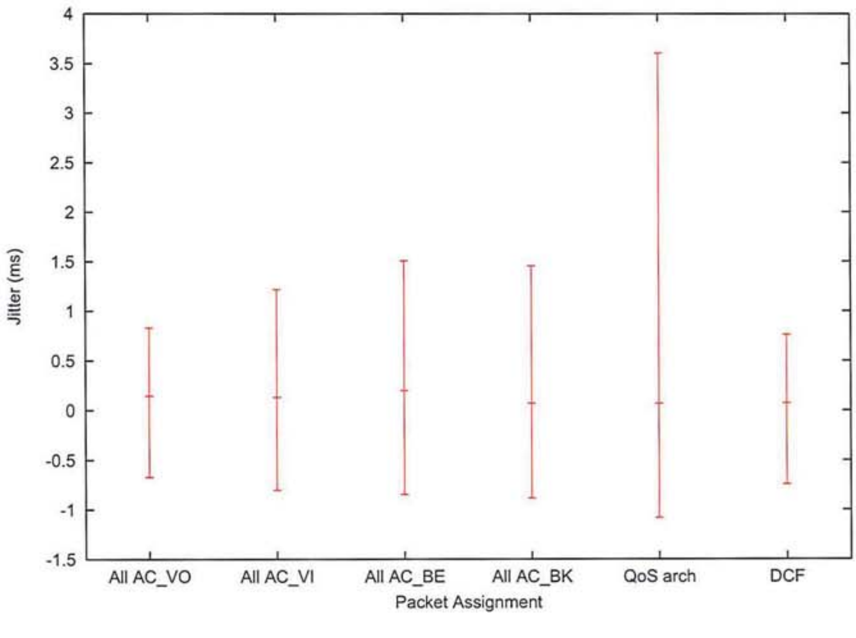


Figure 5.27: Average Jitter with one standard deviation either side

	IDR	Partition A	Partition B	Partition C
AC_VO	100.000%	99.897%	99.962%	99.720%
AC_VI	100.000%	99.944%	99.957%	99.710%
AC_BE	99.171%	99.277%	98.656%	99.570%
AC_BK	99.171%	99.803%	99.726%	99.715%
QoS arch	99.171%	99.122%	81.369%	87.312%
DCF	98.673%	99.122%	98.905%	99.092%

Table 5.7: Percentage of packets received by the client

	IDR	Partition A	Partition B	Partition C
AC_VO	0.000%	0.075%	0.024%	0.251%
AC_VI	0.000%	0.042%	0.029%	0.275%
AC_BE	0.332%	0.113%	0.250%	0.092%
AC_BK	0.663%	0.146%	0.187%	0.232%
QoS arch	0.337%	0.150%	18.525%	12.635%
DCF	0.000%	0.005%	0.005%	0.010%

Table 5.8: Percentage of packets lost in wireless channel

Table 5.7 shows the percentage of packets received by the application at the video client. In Tables 5.8 and 5.9 we highlight the percentage of packets lost in the wireless channel and those dropped by the access point respectively. From this it can be seen that in some circumstances a greater percentage of packets are lost due to virtual contention at the access point than in the transmission from the access point to the client. In the cases where all the packets are sent in different classes the largest loss of packets occurs in the wireless network and not due to virtual contention at the access point. We also see in Figure 5.26 that when the video is allocated to AC_VO or AC_VI the video quality is better than when using the QoS arch assignment.

Figure 5.27 shows the jitter received by the video stream. The jitter values are within the range of $\pm$2ms. The smallest jitter is experienced when the video is streamed in the AC_VO

	IDR	Partition A	Partition B	Partition C
AC_VO	0.000%	0.028%	0.014%	0.029%
AC_VI	0.000%	0.014%	0.014%	0.015%
AC_BE	0.497%	0.610%	1.094%	0.338%
AC_BK	0.166%	0.051%	0.087%	0.053%
QoS arch	0.492%	0.728%	0.106%	0.053%
DCF	1.327%	0.873%	1.090%	0.898%

Table 5.9: Percentage of packets lost by access point

class. Figure 5.28 shows that AC_VO requires more retransmissions than the AC_VI case, but, as we have already seen, it has the smallest jitter. The reason that AC_VO has both a smaller jitter and more retransmissions, compared to AC_VI, is because it has a shorter contention time between successive attempts to access the medium.

5.3 Conclusion

In our work we investigate the potential effects of path diversity on the quality of the video received by a client. We proposed an assignment scheme to map packets onto transmission paths. Using the presented testbed we show that as the number of paths increases the number of frames experiencing the improved PSNR values increase in some scenarios. We found that in the scenario where network conditions experience little change, path diversity can produce benefits.

We take this further through the introduction of FEC. We find that in the environment where path diversity produces benefits using FEC can further improve the quality of the stream received by the client. This improvement is demonstrated through the increase in the average PSNR and the reduction in deviation from the mean PSNR value. The use of FEC can reduce the proportion of the video stream which experiences errors, from 7% down to 0.45%.

Since the last link to consumers is increasingly becoming a wireless link, we have also investigated the QoS options available within the the IEEE 802.11e standard by streaming

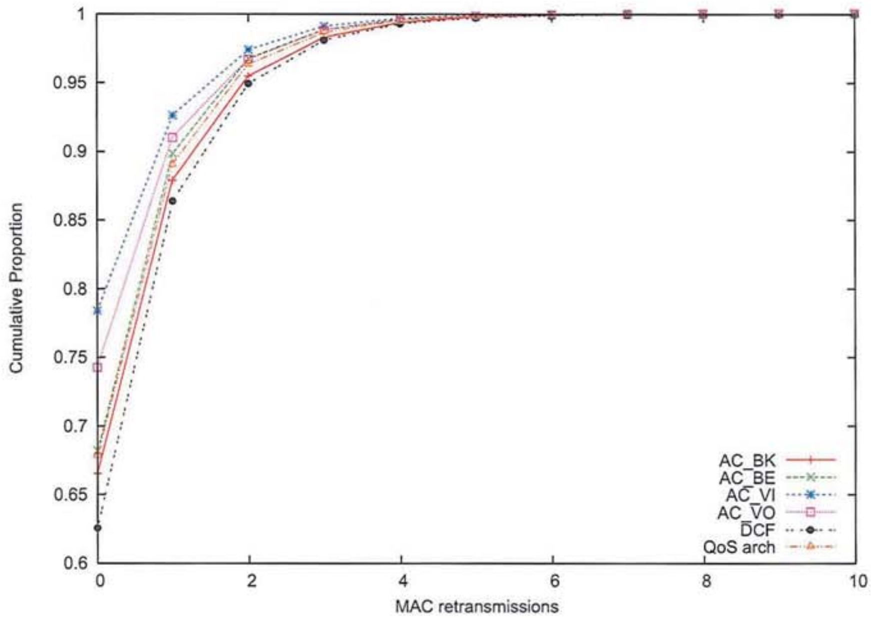


Figure 5.28: Cumulative MAC retransmission

over a real IEEE 802.11e testbed with the use of class assignments. To the best of our knowledge this is the first practical testbed combining 802.11e EDCA with H.264 data partitioning. Over our testbed videos are streamed in each of the EDCA access classes as well as legacy DCF. On top of the video streams we also create best effort TCP traffic to congest the network. From our results we show that assigning video to either AC_VO or AC_VI provides a visible improvement to the received quality at the client, compared to using AC_BE or DCF.

In addition to the experiments for each access category we have also tested the proposed QoS arch class assignment scheme. We found that although QoS arch can provide an improvement over assigning packets to the best effort class, it does not provide any improvement over assigning the video to AC_VO or AC_VI.

In our tests we have seen the effect of virtual contention on the packet stream. In the case of AC_BE and DCF the majority of packet losses occur in virtual contention. For example, with DCF the packet loss rate due to virtual contention is 0.05%, which is much higher than 0.01% for losses in the wireless network.

Chapter 6

Video Caching

To further increase the potential benefits offered by an overlay network we enhance the intelligence at the relay nodes by introducing video caching. Caching provides a number of benefits. Firstly it reduces the processing load, disk access and bandwidth on the original server. Secondly, by having the video stored closer to the client reduces the number of links along which the video has to be streamed. This will result in a lower packet loss rate. ASs generally pay for inter AS traffic, illustrated in Figure 6.1 as B, by caching data in their own network ASs are able to reduce the amount of data they have to stream from other ASs, thus it will reduce costs to ASs [128].

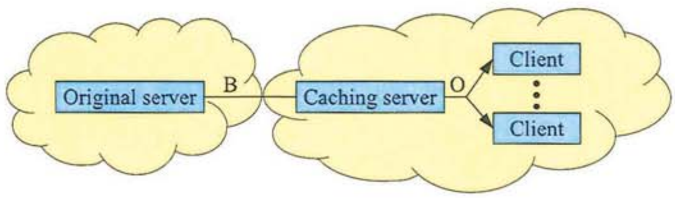


Figure 6.1: Illustration of clients video downloading via a local caching server

As already discussed, videos require large amounts of storage space and larger bandwidths for transmission. Using standard technology to cache large files at the caching server requires that the video be stored in its entirety. Storing the entire video at a cache means a comparatively small number of videos can be stored.

Since not all files are equally popular, some files receive a higher number of requests, putting a greater demand on the server. A number of solutions have been proposed to solve these problems, including popular file caching and prefix caching [129] however no existing caching techniques use data partitioning and store the most important video data near the clients.

By using special video coding techniques it is possible to split a video into a number of streams; each of them can be stored on a different server, thus distributing the load across a number of available caching servers. This has the knock on effect of increasing availability as the resources are used more uniformly and therefore can better cope with the varying demands for different files. The other advantage is that using multiple servers is far more reliable than just using one server which could fail, making the video no longer available.

Caching video close to clients, as shown in Figure 6.1, has a number of benefits. Firstly the reduced latency between the cache and the client will mean that video playback will start quicker. Secondly, caching within the same autonomous system (AS) means that there is likely to be a reduction in the loss rate. This is because it has been shown that 40% of bottlenecked links occur on the links between ASs [130]. Where bottlenecks occur there is a significant increase in the packet loss rate [131].

It is widely agreed that it is not possible to cache video in the same manner as standard web caching, due to its vastly greater file size and long period of time required for delivery [132,133] . A number of different methods have been proposed to cache videos. The authors of [132] propose a system to cache the start of a video stream, resulting in lower response time, server load and network traffic. The authors of [133] proposed a method to cache videos in order to smooth its delivery to the client, which can help to minimise buffer overflows at the client.

The primary goal of the majority of research is to solve the problem of bandwidth or disk usage but not directly the quality of the video received by a user. As the data within an encoded video sequence is not equally important [122], we intend to quantify the value of each video partition and ensure that the more valuable partitions of the video are more likely to be available from a local cache than less valuable partitions.

The authors of [134] investigate the use of layered encoded video and optimise their storage through minimising the blocking probability and thus maximising revenue. Our work differs from theirs in a number of ways. Firstly, we analyse the profit of both the video stored on the caching server and streamed to the client. Secondly, we evaluate the system at the granularity of the partition and not that of the encoded layer. Finally, we use H.264 data partitioning and not a layered codec.

Layered codecs are currently not widely used because of significant loss in coding efficiency and an increase in decoder complexity when compared against a non-layered codec [38]. SVC is one of the most recently developed layered codecs and requires a 10% increase in bandwidth when compared against the single layered H.264 codec for the same fidelity, as measure by PSNR. For the storage of H.264 data partitioned video there is no requirement for an increase in storage space compared to the non-partitioned case. This is because data partitioning is simply a reorganisation of the encoded data and does not result from the compression process itself [37].

Whereas the work presented in [134] looks at a generic layer encoded video, we look at H.264 data partitioning. The work presented in this chapter provides an optimisation of available technology, which could be readily deployed. With H.264 data partitioning we can split the video into a number of sequences of partitions of the same type for frames of the same type, which we term strands. Most strands are dependent on other strands, but this is not a simple layered relationship. For example, for the strand containing partition C of a predicted slice to be decoded, the strand containing partitions A must also be received, however the strand containing partition B could be lost.

Due to the more complex inter strand relationships we are unable to make the assumption that there is a linear improvement in the quality of video as more strands are received. Through the experiments carried out, we investigate the real effect of the loss of a single partition within a strand, which can demonstrate how much improvement we can achieve if this partition has been received.

6.1 PSNR Profit Model

We define PSNR profit, or simply termed profit, as the PSNR difference between the decoded received video and a loss free copy of the video. The summation of the PSNR differences indicates the effect of a loss of a particular partition of a particular frame on the video quality. We thus define PSNR profit of a video as

$$P_V = \sum_{t \in T} \sum_{p \in P} P_{t,p} \times |T_{t,p}| \tag{6.1}$$

where t refers to I, P and B sliced, $T = \{I,P,B\}$, and p refers to the partition type, $P = \{A,B,C\}$. $P_{t,p}$ is the profit of receiving partitions of frame type t and partition type p. To reduce the effect of motion vectors a number of results need to be averaged. Thus the profit, $P_{t,p}$, for a frame and partition pair is defined as:

$$P_{t,p} = \frac{\sum_{r \in T_{t,p}} \sum_{f=1}^{F} \left(PSNR_f - PSNR_{f,r,p}\right)}{|T_{t,p}|} \tag{6.2}$$

where $T_{t,p}$ is a set containing the indexes of the frames of type t containing partition p, where $t \in T$ and $p \in P$. F is the number of frames within the video sequence. $PSNR_f$ refers to the PSNR value of frame f of a compressed video. $PSNR_{f,r,p}$ is the PSNR value of the frame f which has been compressed and has suffered a loss of a partition of type p in frame r. $PSNR_f$ and $PSNR_{f,r,p}$ are defined as follows

$$PSNR_f = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE_f}}\right) \tag{6.3}$$

where MAX_I is the maximum number of encoded levels and MSE_f is the mean square error (MSE) of the compressed image and is defined as

$$MSE_f = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(I_f(i,j) - K_f(i,j)\right)^2 \tag{6.4}$$

where $I_f(i,j)$ is the value of the pixel in the i-th row of the j-th column of the original video frame f and $K_f(i,j)$ is the value of the pixel in the i-th row of the j-th column of the compressed video frame f. m and n are the width and height of the image, respectively.

$$PSNR_{f,r,p} = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE_{f,r,p}}}\right) \tag{6.5}$$

where $MSE_{f,r,p}$ is the MSE for frame f from a video which has lost a partition of type p in frame r.

$$MSE_{f,r,p} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(I_f(i,j) - K_{f,r,p}(i,j)\right)^2 \tag{6.6}$$

where $K_{f,r,p}$ is frame f as a result of H.264 compression and experiencing the loss of partition p in frame r. When data is missing within the frame, recovery techniques are used, which are decoder specific and the motion copy concealment technique [56] is used.

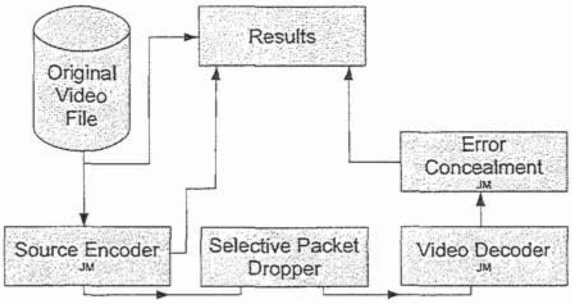


Figure 6.2: Illustration of partition loss test bed

To provide typical profit values we take a ten minute CIF video sequence, from the film Goldeneye. Using the testbed shown in Figure 6.2, we sequentially calculate the profit resulting from the loss of each type of frame and partition. The testbed uses JM v13.2 [121] to encode the video into a sequence of partitions. The GOP size is set to 72 frames, which lasts 3 seconds ($\theta = 3$) when the video is played at 24 frames per second. The GOP is constructed as a key frame followed by alternating bipredicted and intra coded frames. A single partition is removed from the encoded partitions which are then decoded by JM. The PSNR is calculated for the decoded video and then the profit for the single partition is calculated. The averaged PSNR profits are presented in Table 6.1.

	I Slice	P Slice	B Slice
A Partition	1890.63	473.07	9.7
B Partition	509.21	266.91	4.16
C Partition		179.54	1.46

Table 6.1: Average PSNR profit per frame and partition type

6.2 Cache Optimisation

The storage of partitions can be seen as a knapsack problem. Knapsack problems are a type of combinatorial optimization. The aim of a knapsack problem is to maximise the profit stored in a knapsack that can hold a finite number of items. Each item has an associated profit and weight, hence the capacity is fixed in terms of the maximum weight. Knapsack problems can be split into three types, 0-1, bounded and unbounded. In the 0-1 case only one of each item can be carried; in the bounded case there is a finite number of each item which can be carried and in the unbounded case there is an infinite number of each item available to be stored in the knapsack. With regards to H.264 data partitioned video storage, there is a finite number of partitions which can be stored at the cache. Depending on the assignment of profit to these partitions, it can either be represented as a 0-1 knapsack problem or a bounded knapsack problem. We now present more details on both the 0-1 and bounded knapsack problems and solutions.

6.2.1 0-1 Knapsack problem

The 0-1 knapsack problem is to maximise the profit which we can be achieved from the storage of certain components. Each of these components has an associated weight and profit. The target for the knapsack problem is to maximise the profit stored within the knapsack, which has a finite weight , W [135].

There are n components that can be stored in the knapsack, with weight $w_1, ..., w_n$ and a profit $c_1, ..., c_n$, respectively. If we store $\sum_{i=1}^{n} S_i$ components in the knapsack, where $S_i = 0$ or 1 for $i = 1, 2, ...n$. The formal weight bound is that

$$\sum_{i=1}^{n} w_i \times s_i \leq W \tag{6.7}$$

with the target

$$max \left\{ \sum_{i=1}^{n} c_i \times s_i \right\} \tag{6.8}$$

The decision made as to which components should be stored is based on the following algorithm [136]. The components must first be sorted in an order such that

$$\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq ... \geq \frac{c_n}{w_n} \tag{6.9}$$

utilising this ordered sequence, a cut off value is calculated using

$$k = min\left\{ i \in 1,\ldots,n : \sum_{j=1}^{i} w_j > W \right\} \tag{6.10}$$

Then the most valuable $k-1$ components are stored in the knapsack, i.e.

$$\begin{aligned} s_i &= 1 \quad \text{for } i = 1,\ldots,k-1 \\ s_i &= 0 \quad \text{for } i = k,\ldots,n \end{aligned} \tag{6.11}$$

6.2.2 Bounded Knapsack Problem

The bounded knapsack problem is similar to that of the 0-1 knapsack problem and is just a simplification when there are multiple components with the same profit and weight. It would be possibly to use a 0-1 knapsack solution to solve a bounded knapsack problem, however grouping profits and weights can reduce the complexity of the problem.

Like the 0-1 knapsack problem, the bounded knapsack problem is to maximise the profit achieved from the storage of a number of components. These components each have an associated weight, profit and, in the bounded knapsack case, availability, a_i for $i \in 1,\ldots,n$. Again, the components are sorted such that

$$\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq \ldots \geq \frac{c_n}{w_n} \tag{6.12}$$

We store $s_1,\ldots,s_n$ of each component within our knapsack where $0 \leq s_i \leq a_i$ for $i \in 1,\ldots,n$. The decision made as to which components to store is calculated with the following algorithm.

$$s_i = \begin{cases} a_i & \text{if } W - \sum_{j=1}^{i-1} s_j \times w_j \geq a_i \times w_i \\ \left\lfloor \frac{W - \sum_{j=1}^{i-1} s_j \times w_j}{w_i} \right\rfloor & \text{if } W - \sum_{j=1}^{i-1} s_j \times w_j < a_i \times w_i \end{cases} \tag{6.13}$$

The calculation of Equation (6.13) for each value of s_i must be completed in the order $i = 1,\ldots,n$ due to the equations recursive nature. The expression $\sum_{j=1}^{i-1} s_j \times w_j$ is the weight already assigned to the knapsack, thus $W - \sum_{j=1}^{i-1} s_j \times w_j$ is the weight still available for allocation.

6.2.3 Knapsack Solution For H.264 Data Partition Storage

We use the method for calculating profit proposed in Subsection 6.1 and the same video sequence with the values from Table 6.1. In addition to the profit, we also define the weight as the average size of the partitions in bytes, presented in Table 6.2, and the availability as the average number of each partition in a two hour video, i.e. the duration (D) is 7200 seconds. The values for the availability, a_i, of each partition are taken by the summation of the number of the partitions from within the video sequence used. These values are presented in Table 6.3. For the availability used in the calculations the values are multiplied by the number of videos in the system, V.

	I Slice	P Slice	B Slice
A Partition	3466.52	811.36	299.82
B Partition	1294.49	652.08	148.41
C Partition		483.56	177.79

Table 6.2: Average size of a partition in bytes, $W_{t,p}$

	I Slice	P Slice	B Slice
A Partition	1188	85200	86388
B Partition	1188	83304	37716
C Partition		82824	71292

Table 6.3: Average number of partitions, $A_{t,p}$, in a two hour video

	I Slice	P Slice	B Slice
A Partition	0.55	0.58	0.03
B Partition	0.39	0.40	0.02
C Partition		0.37	0.01

Table 6.4: Profit per byte

Using Tables 6.1, 6.2 and 6.3 an optimised number of partitions can be stored on a server, with capacity C bytes. Table 6.4 shows the PSNR profit per byte. The order of the selection for the inclusion of partitions is based on the highest PSNR profit per byte since these partitions provide the best profit for their storage requirements. We define an optimised allocation, $s_{t,p}$ for $t \in T$ and $p \in P$, as one that maximises P within the bounds

$$s_{t,p} \leq A_{t,p} \times V, t \in T p \in P \tag{6.14}$$

$$\sum_{t \in T} \sum_{p \in P} s_{t,p} \times w_{t,p} \leq C \tag{6.15}$$

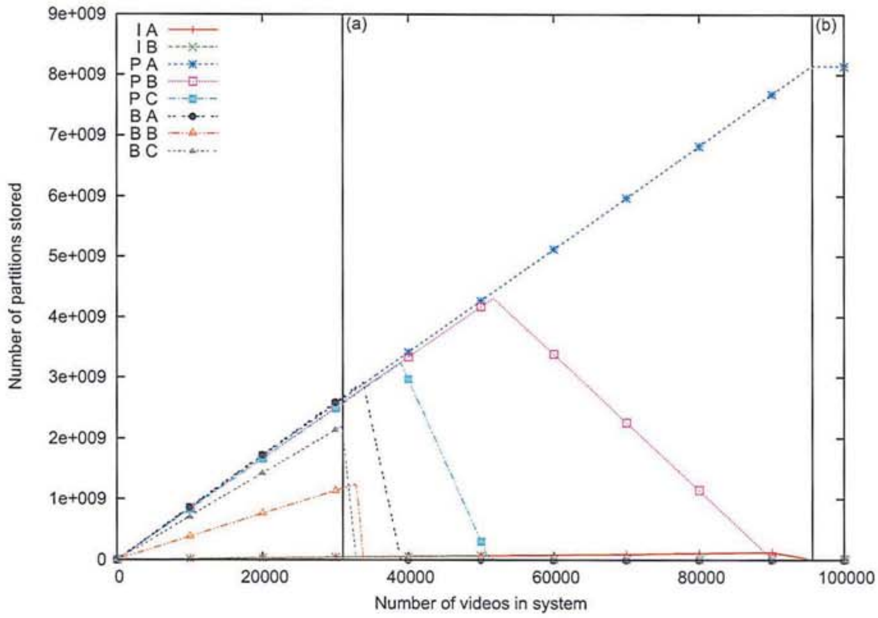


Figure 6.3: Optimised storage of partitions when capacity, $C = 6$TB, while increasing the number of videos in the system, V

Figure 6.3 shows how the optimised number of partitions varies as the number of videos, V, within the system changes, while the server capacity remains constant at $C = 6$TB. In the range $V = 0$ to 31000, marked "(a)" in the figure, the server capacity is not fully used and all partitions are stored. In the range $V = 31000$ to 95600, marked "(b)" in the figure, the server capacity is fully used and the profit of the server increases because less profitable partitions are removed to make space for more profitable partitions. When $V = 95600$ the profitability of the server is maximised. The profitability of the server is defined as

$$P_s = \sum_{t \in T} \sum_{p \in P} s_{t,p} \times P_{t,p} \tag{6.16}$$

While the profit is optimised to maximise the profitability of the components stored at the server, in reality the revenue from a system will be dependent on the profit of the video stream received at the client. On the assumption that each video is equally important we can calculate the average profit of the video stream which is sent from the cache to the client. The profit per video from the cache is defined as

$$P_v = \frac{P_s}{V} \tag{6.17}$$

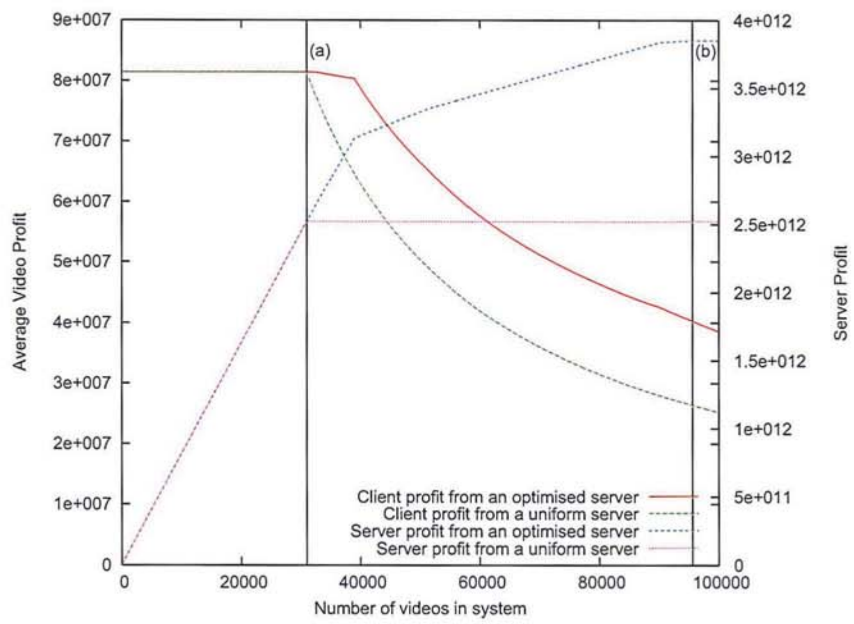


Figure 6.4: Comparison of the profit for both the optimised storage of partitions when capacity, $C = 6$TB, while increasing the number of videos in the system, V

Figure 6.4 shows the average profit for each video stored. The figure compares the optimal and uniform allocation of partitions on the server. In the uniform solution the proportion of storage space assigned to each partition type is the same as the proportions of the amount of space each partition takes up within an average video.

Figure 6.5 shows the probability that all the partitions of a video are available, and that the most profitable partition is available in both the optimised and uniform scenarios. The figure shows that in the optimal case the probability that all partitions of a video are present falls quicker than in the uniform case. The benefits of the optimal system is that the probability that the most important partition of the video is stored is much higher in the optimised case than in the uniform case.

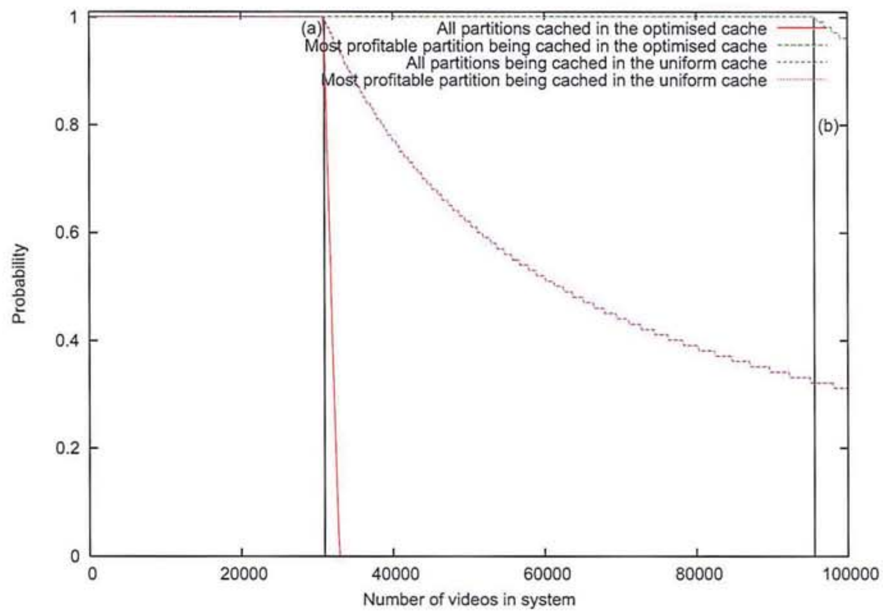


Figure 6.5: Probability that partitions are present within the cache

6.2.4 Caching Network

Within our overlay network we use a number of relay nodes between the server and the client. So far we have only considered a single intelligent caching server. We take this further by enabling clients to stream parts of the video from different caching servers. We define R as the bandwidth required to stream a video to a client and is calculated using

$$R = \frac{\sum_{t\in T}\sum_{p\in P} W_{t,p} \times A_{t,p}}{D} \tag{6.18}$$

The maximum number of clients that a caching server can service is limited by two factors. Firstly, it is limited by the bandwidth between the cache and the clients

$$R \times T \leq O \tag{6.19}$$

where T is the number of target clients and O is the available outbound rate from the caching server. Secondly the number of supported clients is limited by the bandwidth between the original server and the caching server, B, i.e.

$$\frac{\sum_{t\in T}\sum_{p\in P} W_{t,p} \times (A_{t,p} - s_{t,p})}{D} \times T \leq B \tag{6.20}$$

If we make the assumption that the connection between the cache and the client is able to operate at 1 gigabit per second, $O = 1.3 \times 10^8$ bytes per second, and the connection between the cache to the original server is 100 megabits per second, $B = 1.3 \times 10^7$ bytes per second. In reality there will be overheads which will affect the actual streaming rate but these are overlooked in this analysis for simplification. Additionally, we assume that the bottleneck is the network bandwidth and not the disk or server capabilities.

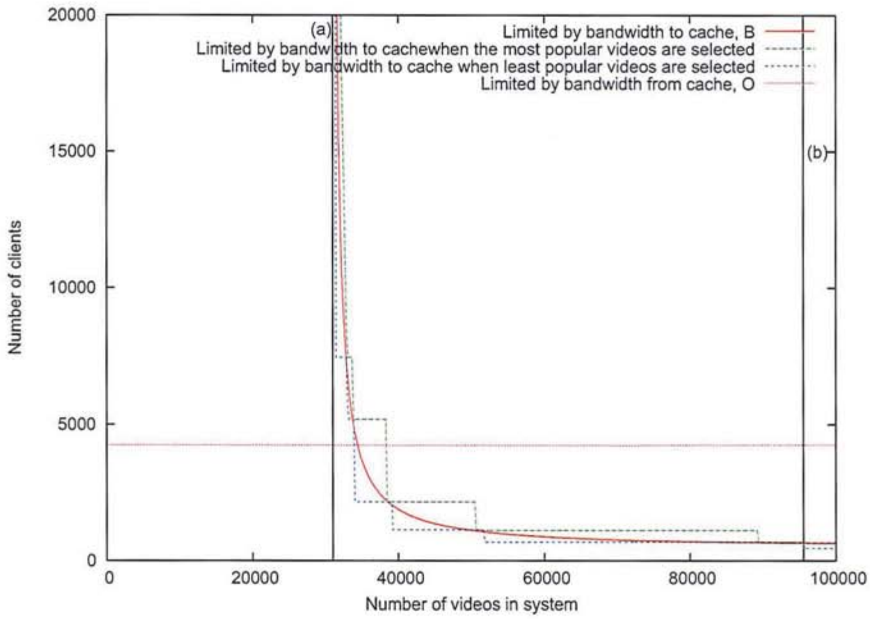


Figure 6.6: Number of clients the system can cope with.

Figure 6.6 shows the number of clients that the system is able to serve complete copies of the video and how this is affected by the number of videos stored at the cache. The number of clients limited by the cache is always constant, as the amount of bandwidth available is finite, calculated using Equation (6.19). The number of clients limited by the bandwidth to the cache is based on the calculation of the amount of additional data that has to be streamed to the cache in order for the cache to stream an entire video to the client, which is calculated using Equation (6.20). In reality this will be dependent on which videos are cached and which videos are requested. In Figure 6.6 we include two limits one where the video requested is the first one to be removed from the cache and the second when the video selected is the last removed from the cache, between these is the limited bandwidth based on the average video selection.

Without the caching server, for the parameter set previously, the number of clients supportable would be limited to 421. When there are 100,000 videos the average number of supported clients is still over 40% higher than without the cache.

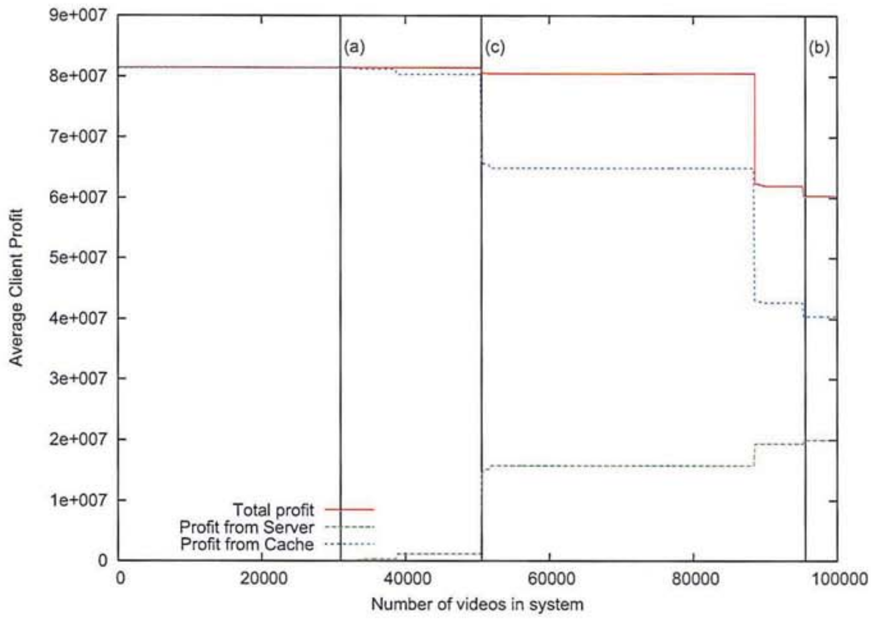


Figure 6.7: PSNR profit when 2000 clients are connected to the cache

Through applying our knapsack optimisation to the bottlenecked bandwidth between the server and the cache, we can provide a more graceful degradation in the quality of the video streamed to the client. Figure 6.7 shows how the PSNR profit degrades when there are 2000 clients each streaming different videos. Figure 6.6 shows that for 2000 clients selecting the most popular videos no more than 50,600 videos can be stored for a perfect copy of the video to be streamed to the client. This point is labelled "C" in Figure 6.7. Figure 6.7 shows how the PSNR profit degrades with an increase in the number of videos stored at the cache. These results should be considered as an upper bound as it assumes that the partitions relating to the videos streamed to the 2000 clients are present at the cache. As the number of videos in the system increases these partitions are removed from the cache last.

Taking this further, we investigate the effect of packet loss on the video profit received by the client. If we assume a uniform distribution of errors, then the profit of the videos streamed from the cache to the client is equal to

$$P_c = \frac{s_{t,p}}{A_{t,p} \times V} \times (1 - L_O) \times P_{t,p} \tag{6.21}$$

where L_O is the packet loss probability of the stream from the cache to the client. The profit of the videos streamed from the server to the client via the cache is equal to

$$P_s = \left(1 - \frac{s_{t,p}}{A_{t,p} \times V}\right) \times (1 - L_O) \times (1 - L_B) \times P_{t,p} \tag{6.22}$$

where L_B is the loss probability of the stream from the server to the cache. The total profit for the streaming of a video is equal to

$$P_t = P_c + P_s \tag{6.23}$$

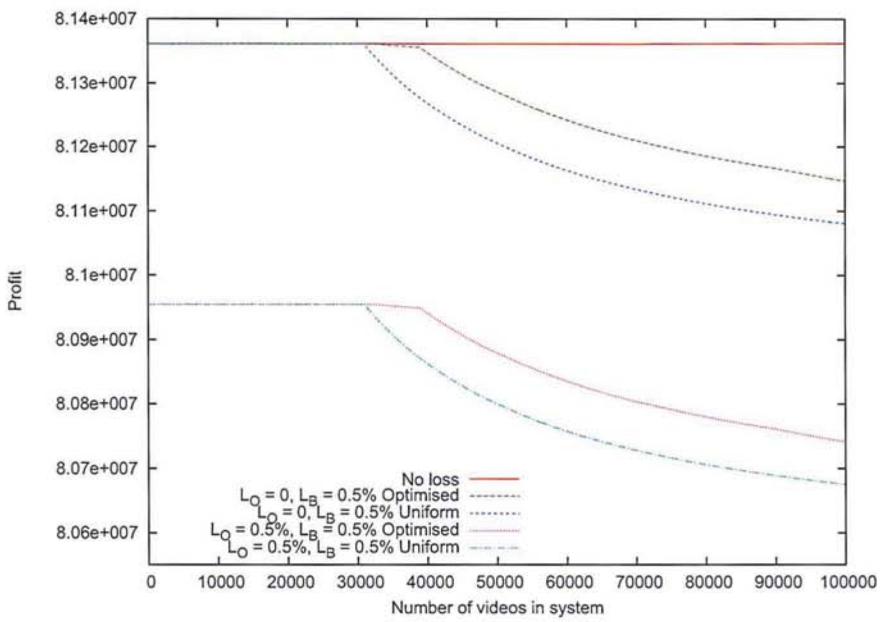


Figure 6.8: The resulting PSNR profit in a lossy environment and how this changes with the number of videos

Figure 6.8 shows the effect that losses have on the resulting PSNR profit received by the client and how this is affected by the number of videos in the system. The number of clients is less than the bound imposed in Figure 6.6. Thus the effects present in Figure 6.7 do not occur.

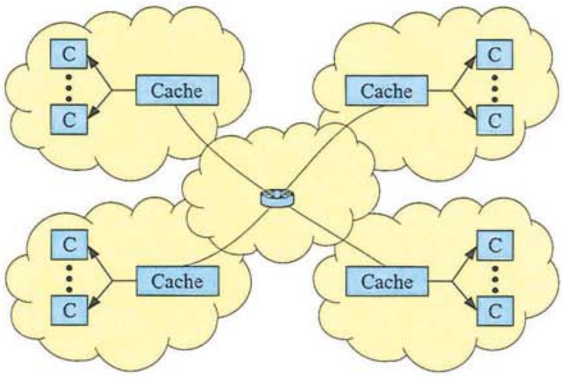


Figure 6.9: Illustration of a caching network

Since the cache is closer to the client than the server, any errors experienced between the cache and the client will also be experienced on the stream from the server. In addition the stream from the server to the cache experiences more errors. Therefore it will always be better to stream the most important parts from the cache. Figure 6.8 shows how the PSNR profit of the client degrades as the number of videos in the system increases, since fewer partitions of the video will be stored at the cache. In all lossy cases the optimised cache outperforms the uniform cache.

6.2.5 Storage Network

If multiple caches exist, potentially within multiple different AS networks as illustrated in Figure 6.9, it is possible that these caches can co-operate to ensure that a copy of the video persists. Within the previously proposed arrangement it would not be possible to ensure that a video remained available as the number of videos within the system grows. Instead of allowing the number of each partition stored on a server to fall to zero, we can either fix a minimum number of partitions to be stored on each server or vary the minimum partitions storable dependent on the number of servers available and the number of videos in the system. By fixing the minimum number of videos completely stored on a server, the number of servers required will vary depending on the number of videos within the system and will be equal to

$$S_{Fixed} = \begin{cases} N & \alpha \geq 0 \\ \frac{V \times N}{F} & \alpha < 0 \end{cases} \tag{6.24}$$

where S_{Fixed} is the number of servers required when the minimum number of videos to be stored on a server is fixed. F is the number of videos which are guaranteed to be stored on each server and N is the number of copies of each video to be stored in the network. Here α is the amount of storage space used on the server but with sufficient space to store all the partitions of the least important partition and frame type for N videos, which is given by

$$\alpha = S + (V - N) \cdot \beta - V \cdot \sum_{t \in T} \sum_{p \in P} A_{t,p} \cdot W_{t,p} \tag{6.25}$$

where β is equal to $A_{t,p} \times W_{t,p}$ for the values of t, p which minimises $\frac{P_{t,p}}{W_{t,p}}$. The alternative is to fix the number of servers, and vary the number of videos which need to be stored on each server, which is given by

$$F_{Variable} = \frac{V \times N}{S} \tag{6.26}$$

where S is the number of servers within the network and $F_{Variable}$ is the number of videos to be stored at a server when the number of servers are fixed and the number of videos can change.

Figure 6.10 presents four scenarios with different N and F parameters. As an example, in the case where $N = 10$ and $F = 10,000$ this means that there will be sufficient space allocated to store 10 complete copies of every video across all the caches, with each cache storing 10,000 complete videos on each server. The storage space required for storing 10,000 complete videos is 32.34% of the complete storage space available.

The results presented in Figure 6.11 distribute N copied of a video over S servers equally. The results show the percentage of storage allocated to the storing of complete videos. The remainder of the storage will is used in the previously proposed optimised proportions.

6.2.6 Partition Selection

The methods presented previously are to optimise the number of each partition type stored at the caching server. However, within the available partitions there is still the questions

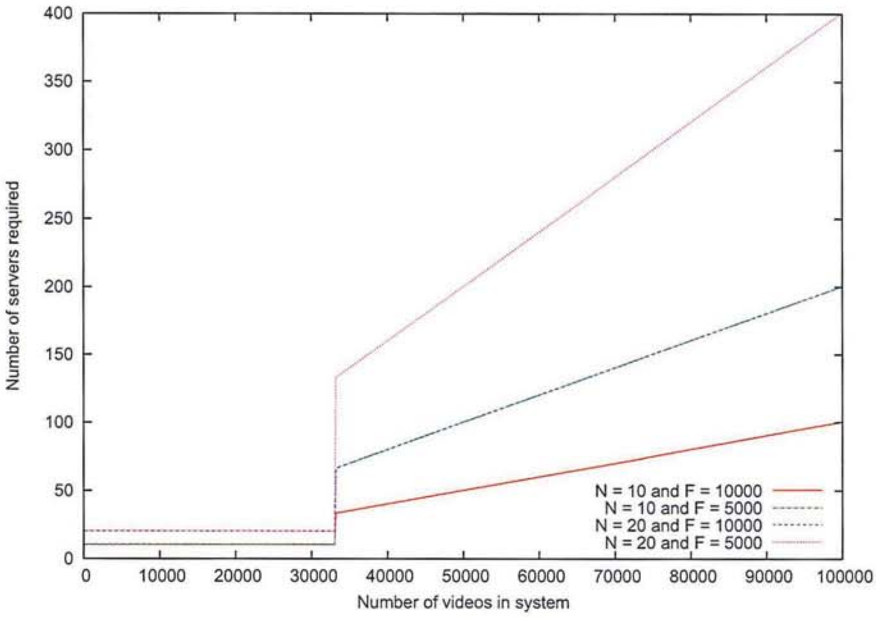


Figure 6.10: Number of servers required to store N copies of all videos when F videos are stored per server fixing the minimum number of videos completely stored

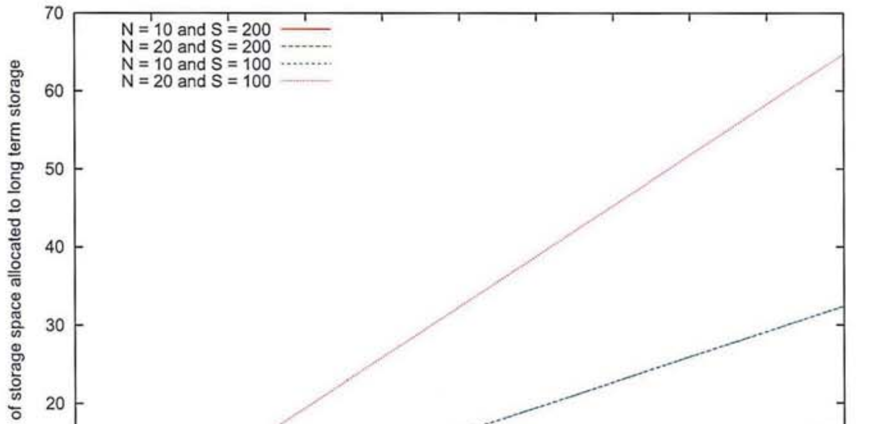


Figure 6.11: Number of servers required to store N copies of all videos when F videos are stored per server varying the number of videos stored on each server

as to which partitions to choose. With the use of data partitioning the strand file sizes are smaller than that of the complete video, but there are multiple strands per video. Since the size of a strand is smaller the removal decision could be made using traditional web caching algorithms such as least recently used (LRU) or least frequently used (LFU) [137]. However, these algorithms fail to take into account video specific improvements.

Further optimisation might be possible through a micro examination of the partitions within the video strands. In our analysis so far we have optimised the number of partitions stored, and not the number of strands. We need to select which partitions from the strands should be stored at the server. One of the targets for a caching algorithm is to require rele-tively little processing. The optimisation can be simply achieved by monitoring the NAL unit octets. There are a couple of other similar optimisation techniques which can be exploited with small processing overheads.

There are two video encoding possibilities, at a constant bit rate and variable bit rate. In the constant bit rate scenario the parameters, such as quantisation, vary to maintain a constant bit rate. In the variable bit rate case the output rate is dependent on the contents of the video and where there is more change a greater bit rate is used, thus the partitions are bigger.

Using the ten minutes of encoded video footage previously discussed encoded, we plot the partition size verses the PSNR to identify if there is any correlation. Figure 6.12 shows the PSNR of the slice which receives the specific loss, against the partition size. From the results the correlation is then calculated using the formula.

$$\frac{\sum_{i=1}^{N}(x_i-\overline{x})(y_i-\overline{y})}{(N-1)s_x s_y} \tag{6.27}$$

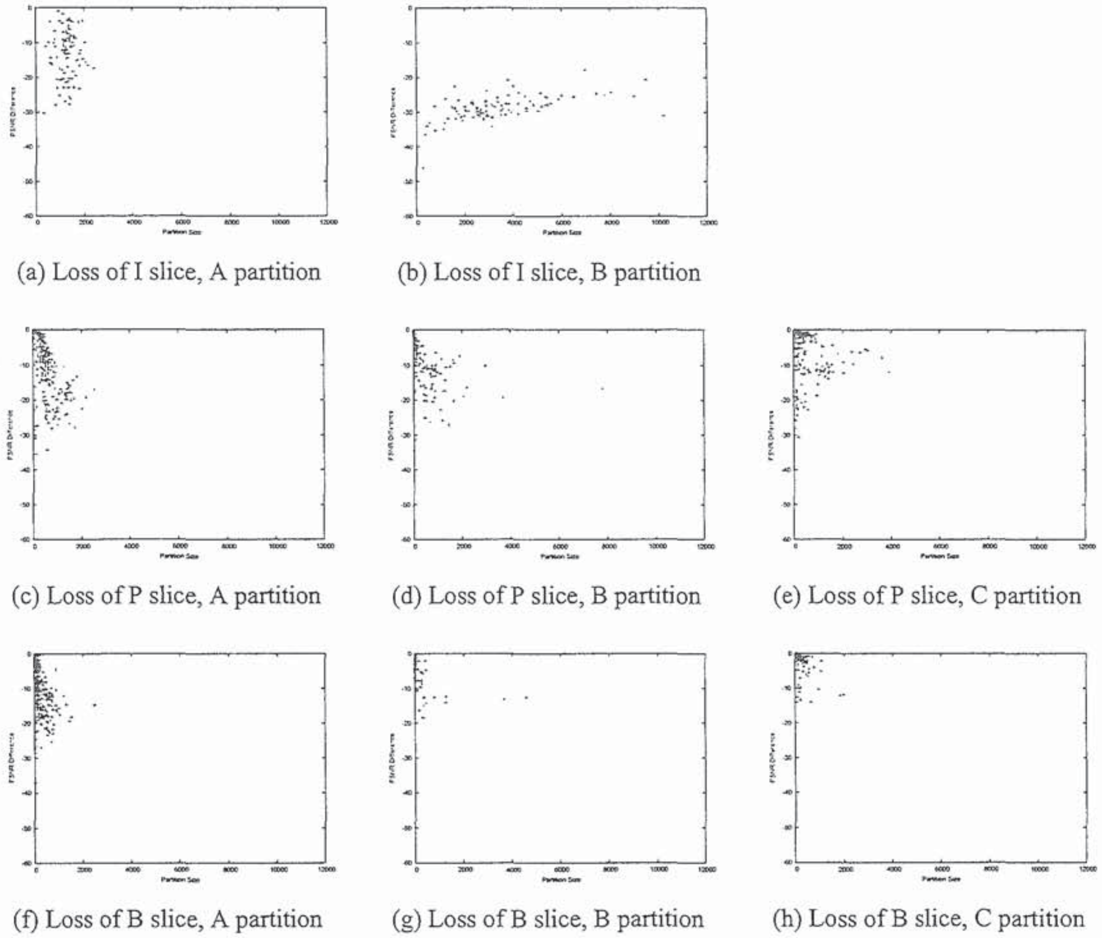
Where x_i and y_i is the location of point i for $i = 1,...,N$, N is the number of data points, $\overline{x}$ and $\overline{y}$ are the average of all x values and y values respectively and s_x and s_y are the standard deviation of x and y respectively.

The corresponding correlation results are presented in Table 6.5.

Table 6.5 shows that there is some correlation between the partition size and the PSNR. For the decoded P and B slices there is a correlation showing that larger packets results in a larger PSNR difference in the decoded slices. In the case of I slices, there is little correlation for A partitions. For the B partitions of I slices there is a near uniform PSNR difference when compared against the partitions size.

	I Slice	P Slice	B Slice
A Partition	0.069	-0.364	-0.273
B Partition	0.559	-0.417	-0.386
C Partition		-0.227	-0.498

Table 6.5: Correlation coefficients for the PSNR difference between the resulting frame and the original against the partition size



(a) Loss of I slice, A partition (b) Loss of I slice, B partition

(c) Loss of P slice, A partition (d) Loss of P slice, B partition (e) Loss of P slice, C partition

(f) Loss of B slice, A partition (g) Loss of B slice, B partition (h) Loss of B slice, C partition

Figure 6.12: The effect on the PSNR difference from the loss of a partition dependent on the partition's size, broken down by frame and partition type

Using the previously presented PSNR profit model we examine the correlation between the PSNR profit and the partition size since this would be a very simple process to deploy on a server.

Using the encoded ten minutes of video footage we plotted the partition size verses the PSNR profit to identify if there is any correlation between them, as shown in Figure 6.13. The correlation coefficients are shown in Table 6.6.

	I Slice	P Slice	B Slice
A Partition	-0.008	0.088	0.274
B Partition	-0.101	0.091	-0.020
C Partition		0.210	0.498

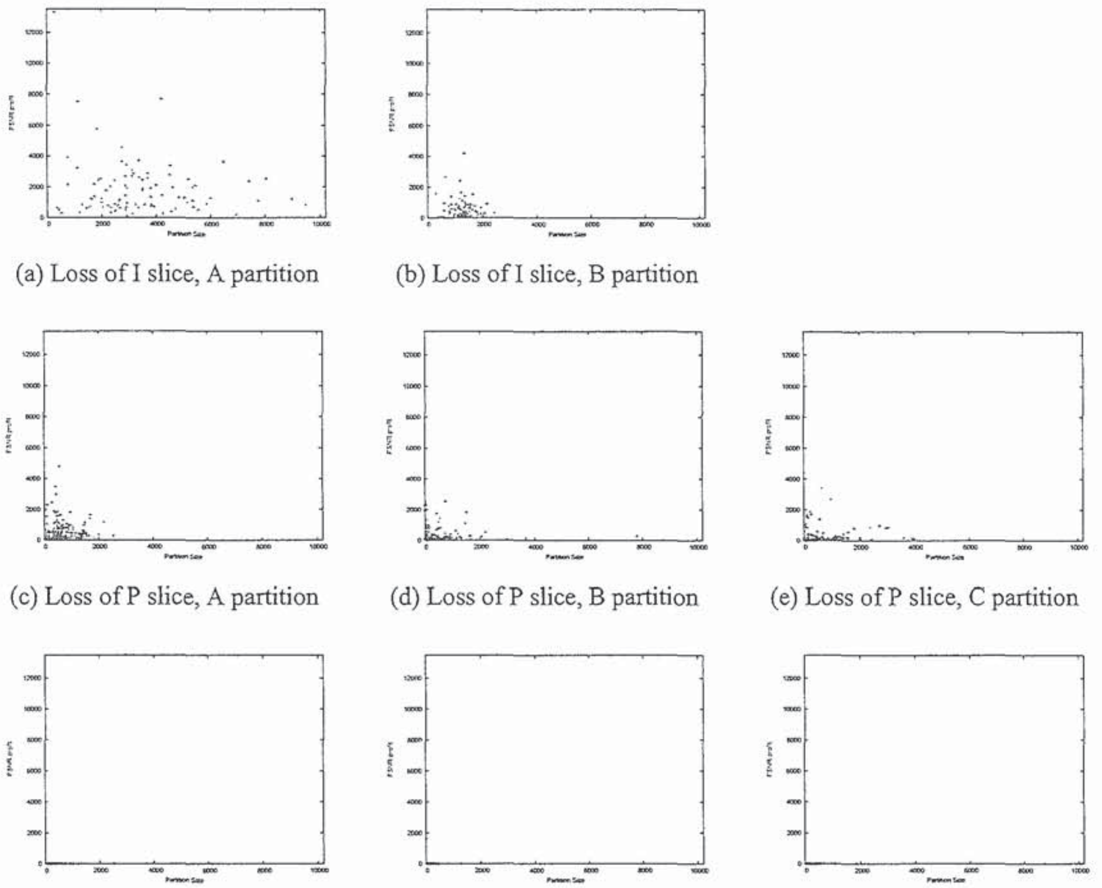
Table 6.6: Correlation coefficients for the PSNR profit against the partition size

Table 6.6 shows that partition size alone is unable to provide an indication of its importance. When considering the partition size and the position within the GOP together improvements could be made to the video quality. We have seen that there is some correlation between the size of a partition and the quality of the decoded version of the frame from which it was decoded, as shown in Table 6.5. It has also been seen that, without data partitioning, frames earlier in a GOP are more important than the later ones [138]. We now look to see if this also applies to data partitioned streams.

Unlike the frame type, partition type and partition size, the frame number or its position within a GOP cannot be identified from the packet itself. To identify the position within a GOP, a counter of the number of A partitions received at the cache is kept. This counter is reset to 0 when an A partition of an I slice is received.

Figures 6.14 shows how the profit is effected by the position of the loss within the GOP. This figure shows the results for P slices and Figure 6.15 shows the results for B slices. From these we see that frames earlier in the GOP have a slightly higher PSNR profit than the frames later in the GOP, this is true both for P and B slices.

Removing partitions throughout the video produces benefits when additional video data is streamed from the server to the cache or client, since the partitions do not need to be buffered for as long before playback. The alternative to this would be to stream the video just before playback but this would result in a higher bandwidth, although over a shorter period of time.



(a) Loss of I slice, A partition

(b) Loss of I slice, B partition

(c) Loss of P slice, A partition

(d) Loss of P slice, B partition

(e) Loss of P slice, C partition

(f) Loss of B slice, A partition

(g) Loss of B slice, B partition

(h) Loss of B slice, C partition

Figure 6.13: The effect on the PSNR profit from the loss of a partition dependent on the partitions size, broken down by frame and partition type

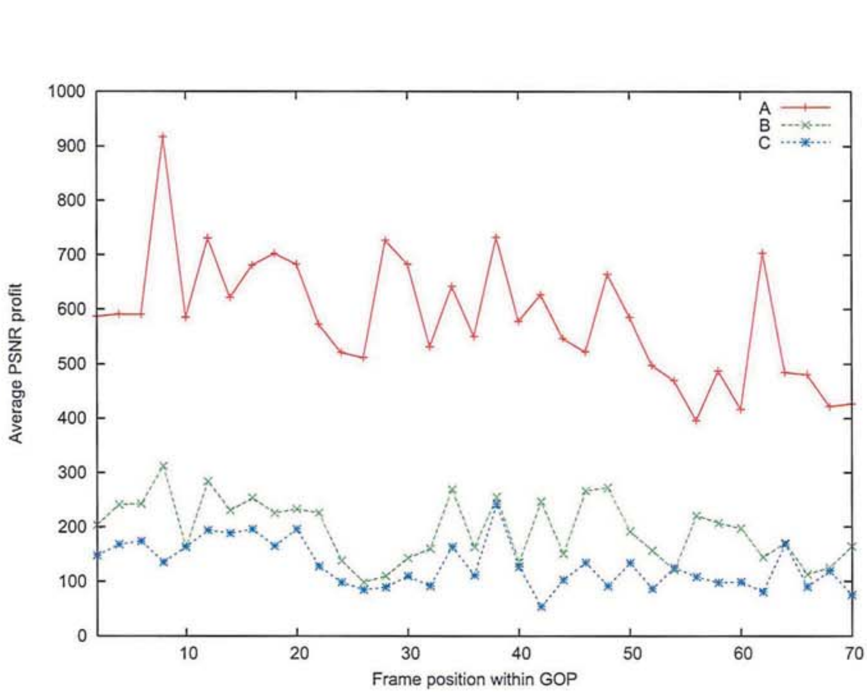


Figure 6.14: The PSNR profit from receiving Intra and Predicted slices

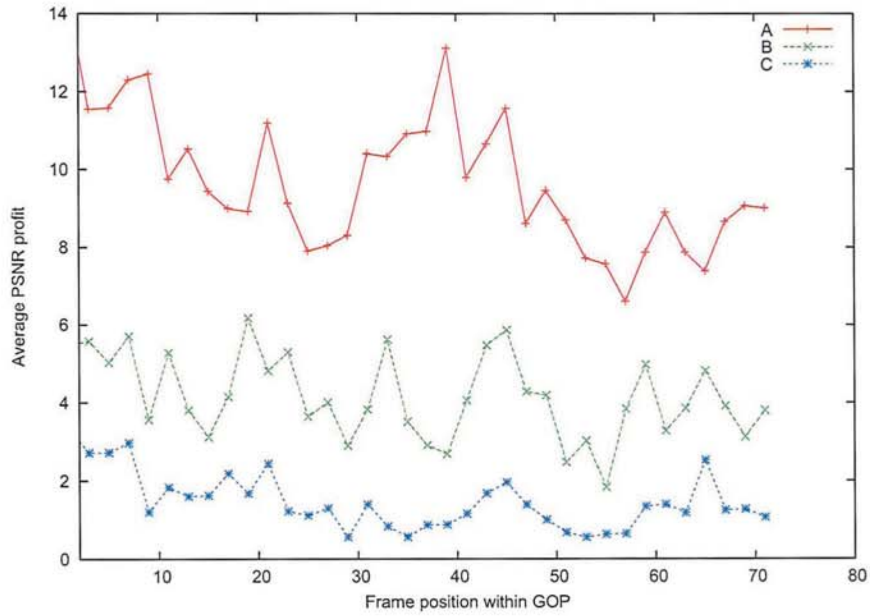


Figure 6.15: The PSNR profit from receiving Bi-predicted slices

For the user to feel that the video on demand system is responsive, video playback should start as soon after the user requests the video as possible. We assume that the cache is closer to the client than the server. The earliest start time is one RTT to the cache plus the duration to receive enough data to start playback, which occurs at time t_0. Figure 6.16 shows a sequence of communications between a client, cache and server. This could be taken further by including archive storage, such as tape, videos requested infrequently could be stored.

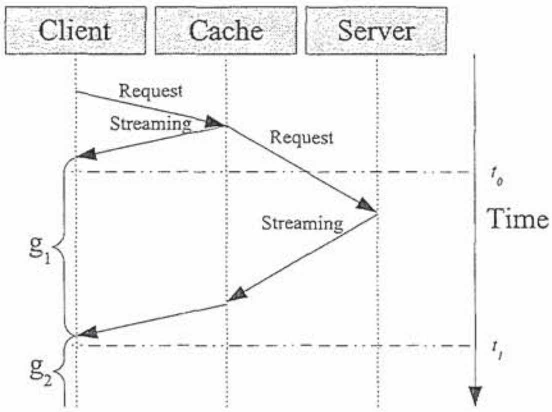


Figure 6.16: Server interaction sequence diagram

It should be highlighted that the RTT is the duration between the request being made and a response to it. In the video scenario the RTT will also include the start of streaming from the hard disk drive (HDD), or when accessing a tape archive this could include job queuing and media access.

To minimise the duration between the client's request and the start of the stream, the video played back in the period between the arrival of the stream from the cache and the start of the reception of the stream from the server, labeled g_1 in Figure 6.16, should be available in the cache. Any video arriving after this, labelled g_2 in Figure 6.16, can either be served from the cache or from the original server while maintaining the same quick start.

Within this section we have highlighted a number of techniques which could be used to select partitions to store or to remove from a cache. We propose that the initial portion of the video, for which playback would need to be paused for the video to arrive from an alternative server, should be stored at the server for as long as possible. Within the rest of the video, partitions should be removed from the end of the GOP selecting partitions of type C, then B and finally A of the B slices first, then P slices and finally I slices. When there are a number of partitions which could be removed we propose to remove the smallest partitions as we have shown that a correlation exists between an individuals frame quality and that of the partition size.

6.2.7 Unequal Popularity Distribution

In the work presented the server is optimised on the assumption that all videos are of equal popularity. In reality this is unlikely to be the case. One way of approximating the popularity of a video is to use the Zipf [139, 140] distribution which is a power law distribution and can be formalised as

$$\frac{f(k;s,N) = 1/k^s}{\sum_{n=1}^{N} \frac{1}{n^s}} \tag{6.28}$$

To incorporate the Zipf distribution we redefine the profit of each partition as

$$P_{t,p,v} = P(v)_{s,M} \times P_{t,p} \tag{6.29}$$

where $P(v)_{s,M}$ is the popularity of video v. In the previous section the popularity of each video, $P(v)_{s,M}$, was 1. In the case of unequal popularity we set the mean value of the video popularity, $P(v)_{s,M}$, to 1. We also define s as a constant which characterises the exponent of the distribution. For the remainder of this analysis we assume that $s = 1$. M is the maximum number of possible videos. $P(v)_{s,M}$ is defined as

$$P(v)_{s,M} = \frac{M}{v^s \times \sum_{n=1}^{M} \frac{1}{n^s}} \tag{6.30}$$

Figure 6.17 shows the number of partitions of each partition type stored at the cache as the number of stored videos increases. It is assumed that the more popular videos are added to the server first. Compared to the case with uniform video popularity, as shown in Figure 6.3, we see that the server becomes optimised with a lower number of videos ($V = 48,100$ shown as "(d)" in the figure).

In Figure 6.18 we show how the proportion of the server storage is allocated to the different partition types once the server has reached its optimised point. In the case where the video popularity is not uniform, the mixture of partition types stored has more variety than

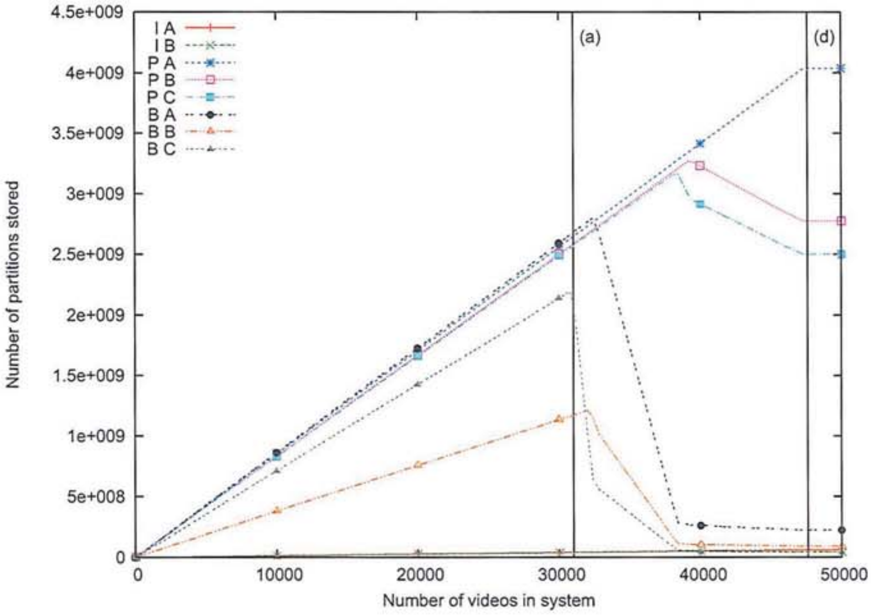


Figure 6.17: Optimised storage of partitions with capacity $C = 6\text{TB}$ while increasing the number of videos in the system, V. Where the videos in the system are not uniformly popular

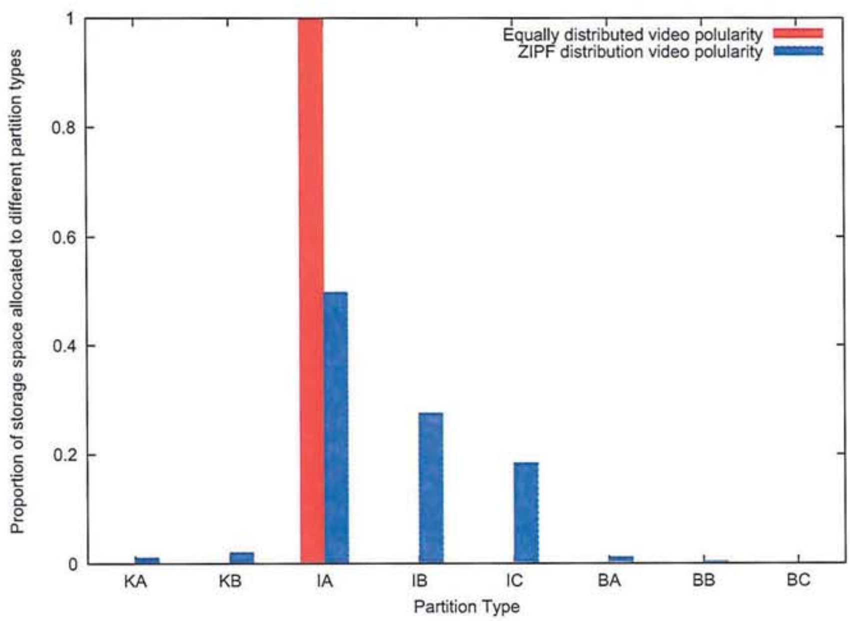


Figure 6.18: The proportion of storage space allocated to different partition types

in the case where the popularity is uniform which results in the storage of a single partition type.

6.3 Conclusion

In this chapter we have presented a way to measure the importance of the partitions of H.264 video using the defined PSNR profit. The typical profit values for our work are generated from real videos and processed using currently available encoding and decoding software. We have then applied an optimisation algorithm to maximise the benefit that an individual cache or a caching network can provide, from both a server's and client's point of view.

From the results obtained, we can see that using the optimal caching scheme proposed the profit of the server keeps rising as the number of videos in the system increases even when the storage has reached it maximum capacity, while in the same situation the profit of conventional caching remains unchanged as soon as the cache becomes full. In addition, we have shown that the profit received by the client through optimal caching will never be less than that of the conventional caching scheme; and when the caching server is fully utilised the former degrades more gracefully than the latter as the number of new videos increases.

We have also shown that the optimal caching strategy can provide benefit when the losses exist in both the server-cache and the cache-client links. In a caching network environment, it has been shown that the optimisation process can ensure the storage of a complete copy of every video within the network.

In our work, we restrict the investigation to the PSNR profit that results from a single partition loss affecting the video stream at any point in time. This greatly simplifies the construction and analysis of the model and this can be extended to address the multiple partition loss cases in the future work.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we initially presented an overview of video source coding, ranging from H.261 up to H.264. H.264 is more efficient than each of its predecessors, when comparing videos with the same fidelity. Within the H.264 standard there is the option for data partitioning, which adds the ability to separate data of different importances into different packets without any loss in compression efficiency.

There are many ways to transport video streams between a server and a client, however we focus on the delivery of video packets over best effort IP networks, such as the Internet. We have discussed different transport protocols, including TCP and UDP, in relation to the streaming of H.264 data partitioned video. Of the two transport protocols, we select UDP because of its flexibility for the transportation of video containing partitions of different importances, which is not offered by TCP.

There are a number of existing QoS frameworks for IP networks, such as IntServ which provides QoS guarantees but is not scalable to an Internet size network. The major alternative to IntServ is DiffServ, which provides a class based QoS architecture but it does not offer service guarantees and is unable to prevent issues resulting from router updates. The major benefit of DiffServ is that it can scale to Internet size environments.

One proposed method to further improve Internet QoS is through the use of path diversity. There are a number of ways to implement path diversity but only multihoming and overlay networks are practical approaches to providing path diversity in real applications. The decision between overlay networks and multihoming is likely to be based on cost and development issues [107], as opposed to technical reasons. Of the two approaches, implementing path diversity in an overlay network is likely to be cheaper since it only requires a connection to a single ISP and a single contract, whereas multihoming requires contracts with multiple ISPs and the related additional infrastructure.

Most researchers investigate path diversity for only two paths. We have extended this to investigate the possible improvement in loss characteristics of a video stream disseminated through more than two paths. This is initially conducted through the establishment of analytical models for random, burst and congestion scenarios. The analytical models are further supported through the use of simulations. The results from both the analytical models and simulations show that in random and burst lossy environments the use of path diversity increases the probability of packet loss for the stream. In the congested scenario, however, there is a reduction in the packet loss probability due to the reduced probability of router buffer over flow as a result of the diverted traffic to the multiple paths used.

By taking these analytical models further, a combined model is presented, to address the real scenarios in the Internet. Using the combined model developed, it is possible to find an optimal number of paths in order to minimise the packet loss rate for the given network conditions. To further investigate path diversity we set up an overlay network on the Internet using PlanetLab. We assumed that the feedback channel is unavailable in our investigation, and that packets are dispatched towards the destination in a round robin manner. The results presented cover the scenarios where up to six paths are selected, based on the criterion that the paths with the fewest number of links are selected among all the paths available in the overlay network. From the results obtained, we see that increasing the number of paths could increase as well as decrease the packet loss probability. There are many factors that contribute to the loss properties concerned. In general, if congestion is the main source of packet losses, path diversity can help divert traffic and reduce the loss probability as a result. However, if non-congestion related losses (random or burst) are dominant, path diversity may lead to more losses.

We have investigated the effect of packet loss patterns on the quality of the video received by the client. We have proposed a mapping scheme of packets onto paths over the testbed presented, and we found that as the number of paths increased the number of frames experiencing the improved PSNR values increased in scenarios where the network conditions are not too changeable. We have also shown that using FEC in conjunction with path diversity can further improve the quality of the stream received by the client. This improvement is in terms of both the increase in the average PSNR and a reduction in the deviation from the mean PSNR value. The use of FEC can reduce the proportion of the video stream which experiences errors from 7% down to 0.45% when 3.2% redundancy is added.

We have shown that when the last link of the network is wireless, path diversity can still be beneficial as long as the backbone of the network experiences congestion losses. We have also investigated the QoS options available within the IEEE 802.11e standard by streaming over a real IEEE 802.11e testbed with the use of class assignments. To the best of our knowledge, this is the first practical testbed combining 802.11e EDCA with H.264 data partitioning, which has been presented at the IEEE International Conference (ICC) in June, 2009. Over the testbed videos are streamed in each of the EDCA access classes as well as legacy DCF and, at the same time, the network is congested by best effort TCP traffic. From our results, we show that assigning video to either AC_ VO or AC_VI provides a visible improvement to the received quality at the client, compared to using AC_BE or DCF. We have evidenced of the effect of virtual contention on the packet stream. For example, with DCF the packet loss rate due to virtual contention is 0.05%, which is much higher than the 0.01% corrupted in the wireless link.

To increase the intelligence of the overlay network on top of providing an intermediate location for packets to be routed through, we have proposed an optimal caching scheme at the intermediate nodes by taking advantage of H.264 data partitioning and creating a way of measuring the importance of the partitions of H.264 video, namely the PSNR profit. The typical profit values for our work are generated from real videos and processed using currently available encoding and decoding software. We have applied an optimisation algorithm to maximise the benefit that an individual cache or a caching network can provide, from both a servers and clients point of view. We have shown that using the optimal caching scheme proposed the profit of the server keeps rising as the number of videos in the system increases even when the storage has reached it maximum capacity, while in the same situation the profit of conventional caching remains unchanged. In addition, we have shown that the profit received by the client through optimal caching will never be less than that of the conventional caching scheme; and when the caching server is fully utilised the former degrades more gracefully than the latter as the number of new videos increases.

7.2 Personal Contribution

In this section we provide an overview of the contributions presented in this thesis.

7.2.1 Analytical Models

We have established the random and burst loss models using the Bernoulli and two state Markov chain models. In addition, a congestion model has also been proposed. These models are applied to the environment where path diversity (with up to 11 paths created) is applied, and each of them are individually compared against simulation results for the validation of the performance obtained. Based on this, a combined loss model is presented to address the situation exhibited in the Internet.

7.2.2 Overlay Network Experiments

We have created overlay networks comprising of multiple paths using PlanetLab, to examine the effect of path diversity on the loss performance in a real-world environment. We have proposed a method to calculate the correlation between different paths in an overlay network. Additionally we proposed an assignment scheme to map partitioned video streams onto the paths available to enhance the video quality experienced by the end user.

7.2.3 Wireless Streaming

We have carried out real video streaming over a WLAN by assigning video partitions to different access classes. We have shown that assigning all packets to the video or voice AC proves more beneficial than a previously proposed cross class assignment scheme. In addition, we have highlighted the issue of virtual contention at the access point and identified the loss property due to virtual contention, compared to the conventional data corruption rate in the wireless medium.

7.2.4 Partition Profit Measure and Optimal Video Caching

To address the diverse importance of different video partitions, we have introduced a measure, termed PSNR profit. Using this measure we have proposed a scheme to optimise the storage of partitions within a cache for H.264 video. This scheme can substantially enhance the robustness and efficiency of a video cache when used in large-scale distributed network and in a lossy environment.

7.3 Future Work

Future work includes developing the loss models, the results provide are based on the approximated parameter values for each of the models used. The work could be further developed by refining these parameters to increase the accuracy of this model.

The results obtained from PlanetLab are taken from three separate overlay networks. For improving the quality of analysis, the number of overlay networks could be increased. It would also be useful to identify if the overlay networks can operate more effectively over smaller or larger geographical areas.

The investigation into a cross class video assignment scheme operated in wireless networks only considers a single video client although with multiple wireless background traffic streams. In this scenario, we find that the cross class video assignment scheme provided no benefits to the client. However, the work could be extended by increasing the number of wireless video receiving client to investigate if the aggregated quality of the video received can be improved across all clients.

For the partition analysis which resulted in the PSNR profit values, we have only investigated the effect of a single partition loss. The work can be taken further to study the effect of receiving different partition loss patterns within a GOP. From this, a more accurate model of mapping partitions to PSNR profit can be devised.

References

[1] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP payload format for H.264 video," Internet Engineering Task Force, RFC 3984, Feb. 2005.

[2] IEEE-SA, "IEEE 802.11e : Amendment: Medium access control (MAC) quality of service (QOS) enhancements," Tech. Rep., 2005.

[3] T. Porter and X.-H. Peng, "Adapting FEC to suit an application when combining RS coding with TCP," Aston University, Tech. Rep., 2009.

[4] J. Postel, "RFC 791: Internet Protocol," Sept. 1981.

[5] K. Ramakrishnan, S. Floyd, and D. Black, "RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP," September 2001.

[6] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "The Case for Resilient Overlay Networks," in *8th Workshop on Hot Topics in Operating Systems*, Elmau/Oberbayern, Germany, May 2001.

[7] IEEE-SA, "IEEE 802.11 : Wireless lan medium access control (MAC) and physical layer (PHY) specifications," Tech. Rep., 1999.

[8] "Youtube," [Online; accessed 14-October-2006]. Available: http://www.youtube.com

[9] R. MacManus, "Youtube may be the fastest growing website ever!" *ZDNet Web 2.0 Explorer*, October 2006. Available: http://blogs.zdnet.com/web2explorer/?p=284

[10] BBC, "BBC iPlayer growth continues as bbc.co.uk records 29% increase," Press Release, Feb 2008, [Online; accessed 5-March-2008]. Available: http://www.bbc.co.uk/pressoffice/pressreleases/stories/2008/02_february/

[11] "Surge in demand for online video," *BBC News*, March 2009, [Online; accessed 20-March-2009]. Available: http://news.bbc.co.uk/1/hi/technology/7954941.stm

[12] P. Cashmore, "Facebook video launches: Youtube beware!" Online, May 2007, [Online; accessed 27-November-2008]. Available: http://mashable.com/2007/05/24/facebook-video-launches/

[13] Facebook, "Facebook video," 2009, [Online; accessed 24-June-2009]. Available: http://www.facebook.com/apps/application.php?id=2392950137

[14] "Myspace video," [Online; accessed 27-November-2008]. Available: http://www.myspace.com

[15] T. O. TV, "Peer-to-peer," 2009, [Online; accessed 16-October-2005]. Available: http://www.timesonline.co.uk/tol/news/times_online_tv/

[16] CNET, "CNET Video," 2009, [Online; accessed 24-June-2009]. Available: http://videos.cnet.co.uk/

[17] "Video on-demand: Behaviour, challenges and future direction," Coda Research Consultancy, Tech. Rep., 2009.

[18] T. Chen, "Video coding standards for multimedia communication:H.261, H.263, and beyond," Carnegie Mellon University, Tech. Rep., 1999.

[19] ITU-T, "H.261 : Video codec for audiovisual services at Px64 kbps," March 1993.

[20] H. Benoit, *Digital Television MPEG-1, MPEG-2 and Principles of the DVB system*. Arnold, 1997.

[21] I. 11172-2, *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video*, 1993.

[22] ITU-T, "H.262 : Generic coding of moving pictures and associated audio information," July 1995.

[23] ——, "H.263 : Video coding for low bitrate communication," March 1996.

[24] ——, "H.263v2 : Video coding for low bitrate communication (version 2)," February 1998.

[25] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Transactions On Circuits and Systems For Video Technology*, vol. 8, pp. 849–866, 1998.

[26] ITU-T, "H.263v3 : Video coding for low bitrate communication (version 3)," November 2000.

[27] Y.-H. Chen, *ITU-T video coding standards*, Data Compression and Multimedia Communication Lab, National Cheng Kung University, Taiwan, [Online; accessed 20-March-2009]. Available: http://dcmc.ee.ncku.edu.tw/pdf/course/MC/MC08.pdf

[28] ITU-T, "H.264 : Advanced video coding for generic audiovisual services," May 2003.

[29] K. Jack, *Video Demystified*, 4th ed., ser. Demystifying Technology Series. Newnes Elsevier, 2005.

[30] M. Nilsson, *Current Status of H.263 and H.26L*, BT Exact Technologies, 2002, [Online; accessed 11-March-2008]. Available: http://www.iee.org/oncomms/pn/visualinformation/Current_Status_of_H.263_and_H.pdf

[31] N. Kamaci and Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards," *In proceedings IEEE International Conference on Multimedia and Expo*, 2003.

[32] A. Luthra, G. Sullivan, and T. Wiegand, "Introduction to the special issue on the H.264/AVC video coding standard," *IEEE Transactions On Circuits and Systems For Video Technology*, vol. 13, no. 7, pp. 557–559, July 2003.

[33] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits and Systems*, pp. 7–28, 2004.

[34] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Transactions On Circuits and Systems For Video Technology*, vol. 13, no. 7, pp. 704–716, July 2003.

[35] A. Rodriquez, J. Garell, J. Karas, and R. Peschkė, *TCP/IP Tutorial and Technical Overview*, 7th ed., ser. RedBooks. IBM, August 2001.

[36] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions On Circuits and Systems For Video Technology*, vol. 13, pp. 560–576, July 2003.

[37] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. Wiley, 2003.

[38] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions On Circuits and Systems For Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

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

[40] R. Bernardini, M. Durigon, R. rinaldo, P. Zontone, and A. Vitali, "Real-time multiple description video streaming over QOS-based wireless networks," in *IEEE International Conference on Image Processing, 2007*, October 2007, pp. 245–248.

[41] T. Stockhammer, M. Hannuksela, and S. Wenger, "H.26L/JVT coding network abstraction layer and ip-based transport," in *Proceedings of the International Conference on Image Processing*, vol. 2, 2002, pp. II–485–II–488 vol.2.

[42] S. Ha, I. Rhee, and L. Xu, "Cubic: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating System Review*, vol. 42, no. 5, pp. 64–74, July 2008.

[43] D. Leith, R. Shorten, and G. McCullagh, "Experimental evaluation of cubic TCP," in *Proceedings of Protocols for Fast Long Distance Networks*, 2007, pp. 7–12.

[44] J. Postel, "RFC 768: User datagram protocol," Aug. 1980, status: STANDARD.

[45] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 1889: RTP: A transport protocol for real-time applications," Jan. 1996, status: PROPOSED STANDARD.

[46] Network Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 3550: RTP: A transport protocol for real-time applications," July 2003.

[47] K. Jeacle, "TCP-xm," Ph.D. dissertation, University of Cambridge, 2005.

[48] A. Hu, "Video-on-demand broadcasting protocols: a comprehensive study," in *INFOCOM*, 2001, pp. 508–517.

[49] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," in *In Proceedings of S&T/SPIE Conference on Multimedia Computing and Networking (MMCN)*. IEEE Press, 1995, pp. 66–77.

[50] K. A. Hua, Y. Cai, and S. Sheu, "Patching: a multicast technique for true video-on-demand services," in *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 1998, pp. 191–200.

[51] H. Eriksson, "Mbone: the multicast backbone," *Commun. ACM*, vol. 37, no. 8, pp. 54–60, 1994.

[52] S. Wolf, M. H. Pinson, A. A. Webster, G. W. Cermak, and E. P. Tweedy, "Objective and subjective measures of MPEG video quality," in *Proceedings of SMPTE Technical Conference*, 1997, pp. 21–24.

[53] P. Schallauer, M. Donoser, G. Kienast, and H. Rehatschek, "High quality compression for film and video," PRESTO, Tech. Rep., 2002.

[54] S. K. Bandyopadhyay, Z. Wu, P. Pandit, and J. M. Boyce, "An error concealment scheme for entire frame losses for H.264/AVC," in *IEEE Sarnoff Symposium*, March 2006, pp. 1–4.

[55] W. Lee, M. Frater, M. Pickering, and J. Arnold, "Spatial temporal concealment of lost blocks in coded video," in *International Conference on Image Processing*, vol. 3, Oct 1998, pp. 477–481.

[56] Z. Wu and J. M. Boyce, "An error concealment scheme for entire frame losses based on H.264/AVC," in *IEEE International Symposium on Circuits and Systems, 2006*, September 2006, pp. 4463–4466.

[57] W. Zhao, D. Olshefski, and H. Schulzrinne, "Internet quality of service: an overview," Columbia University, Tech. Rep., 2000.

[58] M. Welzl, *Network congestion control*. Wiley, 2005.

[59] H. Zimmermann, "OSI reference model–the iso model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

[60] G. Held, *Ethernet Networks*, 4th ed. John Wiley & Sons, LTD, 2003.

[61] J. M. Boyce and R. D. Gaglianello, "Packet loss effects on MPEG video sent over the public internet," in *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 1998, pp. 181–190.

[62] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC 2205: Resource reservation protocol (RSVP)," September 1997.

[63] *Internetworking Technology Handbook.* Cisco, 2008.

[64] J. Wroclawski, "RFC 2211: Specification of the controlled-load network element," September 1997.

[65] S. Shenker, C. Partridge, and R. Guerin, "RFC 2212: Specification of guaranteed quality of service," September 1997.

[66] K. Nichols, S. Blake, F. Baker, and D. Black, "RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," December 1998.

[67] J. Babiarz, K. Chan, and F. Baker, "RFC 4594 : Configuration guidelines for DiffServ service classes," August 2006.

[68] S. Mukherjee, "Quality of service (QOS) over 802.11 wireless networks," Ph.D. dissertation, Aston University, 2009.

[69] D. Clark, "The design philosophy of the darpa internet protocols," in *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols.* New York, NY, USA: ACM, 1988, pp. 106–114.

[70] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "Characterizing and measuring path diversity of internet topologies," in *SIGMETRICS '03: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems.* New York, NY, USA: ACM, 2003, pp. 304–305.

[71] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," *IEEE Communications*, vol. 42, pp. 80–87, Aug. 2004.

[72] Network Working Group, Y. Rekhter, and T. Li, "RFC 1771: A border gateway protocol 4 (BGP-4)," March 1995, status: STANDARD.

[73] J. Stewart, *BGP4: Inter-Domain Routing in the Internet.* Pearson Professional Education, 1999.

[74] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 232–243, 2002.

[75] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?!: it must be bgp," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 75–84, 2007.

[76] Y. J. Liang, E. G. Steinbach, and B. Girod, "Multi-stream voice over IP using packet path diversity," in *IEEE Fourth Workshop on Multimedia Signal Processing*, 2001.

[77] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "1-800-overlays: using overlay networks to improve VoIP quality," in *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video.* New York, NY, USA: ACM, 2005, pp. 51–56.

[78] R. K. Rajendran, S. Ganguly, R. Izmailov, and D. Rubenstein, "Performance optimization of VoIP using an overlay network," NEC Laboratories, Tech. Rep., 2005.

[79] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," in *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication.* New York, NY, USA: ACM, 2000, pp. 175–187.

[80] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz, "Route flap damping exacerbates internet routing convergence," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 221–233, 2002.

[81] D. Magoni and M. Hoerdt, "Internet core topology mapping and analysis," *Computer Communications*, vol. 28, no. 5, pp. 494 – 506, 2005. Available: http://www.sciencedirect.com/science/article/B6TYP-4DGT6XD-3/2/d28dc259207c93111c4a8e37bb3cfd98

[82] M. Hoerdt and D. Magoni, "Completeness of the internet core topology collected by a fast mapping software," in *11th IEEE International Conference on Software, Telecommunications and Computer Networks (softcom)*, Split (Croatie), october 2003.

[83] S. M. Bellovin, "Security problems in the TCP/ip protocol suite," *SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 2, pp. 32–48, 1989.

[84] K. Argyraki and D. R. Cheriton, "Loose source routing as a mechanism for traffic policies," in *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture.* New York, NY, USA: ACM, 2004, pp. 57–64.

[85] S. M. Bellovin, "A look back at "security problems in the TCP/ip protocol suite"," in *ACSAC '04: Proceedings of the 20th Annual Computer Security Applications Conference.* Washington, DC, USA: IEEE Computer Society, 2004, pp. 229–249.

[86] W. Xu and J. Rexford, "Miro: multi-path interdomain routing," in *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications.* New York, NY, USA: ACM, 2006, pp. 171–182.

[87] L. Han, J. Wang, and C. Wang, "A cross-layer concurrent multi-path forward algorithm," in *International Conference on Advanced Infocom Technology*, July 2008, pp. 137–140.

[88] D. Zhu, M. Gritter, and D. R. Cheriton, "Feedback based routing," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 71–76, 2003.

[89] B. Raghavan and A. C. Snoeren, "A system for authenticated policy-compliant routing," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications.* New York, NY, USA: ACM, 2004, pp. 167–178.

[90] H. T. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi, "Bananas: an evolutionary framework for explicit and multipath routing in the internet," in *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*. New York, NY, USA: ACM, 2003, pp. 277–288.

[91] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocols," in *SIGCOMM '87: Proceedings of the ACM workshop on Frontiers in computer communications technology*. New York, NY, USA: ACM, 1988, pp. 2–7.

[92] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2003, pp. 353–364.

[93] A. Akella, S. Seshan, and A. Shaikh, "Multihoming performance benefits: An experimental evaluation of practical enterprise strategies," in *USENIX Annual Technical Conference, General Track*, 2004, pp. 113–126.

[94] J. Balam and J. D. Gibson, "Path diversity and multiple descriptions with rate dependent packet losses," *Information Theory and Applications Workshop*, February 2006.

[95] A. Begen, Y. Altunbasak, and O. Ergun, "Multi-path selection for multiple description encoded video streaming," in *IEEE International Conference on Communication*, vol. 3, 2003, pp. 1583–1589.

[96] A. C. Begen, Y. Altunbasak, O. Ergun, and M. H. Ammar, "multi-path selection for multiple description video streaming over overlay networks," *Signal Processing: Image Communication*, vol. 20, pp. 39–60, 2005.

[97] T. Nguyen, P. Mehra, and A. Zakhor, "Path diversity and bandwidth allocation for multimedia streaming," in *IEEE International Conference on Multimedia and Expo*, 2003.

[98] T. Nguyen and S.-C. Cheung, "multimedia streaming using multiple TCP connections," in *International Performance Computing and Communications Conference*. IEEE IPCCC 2005, April 2005.

[99] S. Wu and S. Banerjee, "An area-based feedback implOSIon control mechanism with deterministic timeouts," in *IEEE Global Telecommunications Conference*, 2001, pp. 1693–1697.

[100] IEEE-SA, "IEEE 802.16 : Air interface for fixed broadband wireless access systems."

[101] 3GPP, "3GPP release 5," 2003.

[102] IEEE-SA, "IEEE 802.3 : Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications," 2005.

[103] ——, "IEEE 802.16e : Amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands," 2006.

[104] Wi-Fi Alliance, "Wi-Fi chipset sales grew 26 percent to 387 million in 2008," Online, January 2009, [Online; accessed 02-February-2009]. Available: http://www.Wi-Fi.org/pressroom_overview.php?newsid=770

[105] IEEE-SA, "IEEE 802.1d : Local and metropolitan are networks media access control (MAC) bridges specifications."

[106] S. Nandagopalan, S. Abraham, and S. Nanda, "How to tune aifs parameters to achieve desired flow ratios or to do congestion/flow control?" Qualcomm, Tech. Rep., May 2006.

[107] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A comparison of overlay routing and multihoming route control," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 93–106, 2004.

[108] S. Floyd and V. Jacobson, "Traffic phase effects in packet-switched gateways," *Journal of Internetworking:Practice and Experience*, vol. 3, no. 3, pp. 115–156, September 1992.

[109] H.-y. Tyan, "J-Sim," Electrical Engineering, Ohio State University, Tech. Rep., 2003, [Online; accessed 22-March-2006]. Available: http://www.J-Sim.org/

[110] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *IEEE INFOCOM*, Apr. 2003, pp. 663–672.

[111] D. D. Clark, "RFC 817: Modularity and efficiency in protocol implementation," July 1982.

[112] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal topology generation from a user's perspective," Boston University, Tech. Rep. 2001-003, 1 2001.

[113] K. L. Calvert, M. B. Doar, and E. W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, June 1997.

[114] K. Fall and K. Varadhan, *The ns Manual*, Nov. 2005.

[115] R. Mortier, "ns-2 introduction," Dec. 2005.

[116] "Opnet modeler," [Online; accessed 23-March-2006]. Available: http://www.opnet.com/

[117] H.-y. Tyan, "Evaluation of J-Sim," Electrical Engineering, Ohio State University, Tech. Rep., 2003, [Online; accessed 23-February-2006]. Available: http://www.J-Sim.org/comparison.html

[118] M. Yajnik, S. B. Moon, J. F. Kurose, and D. F. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *INFOCOM (1)*, 1999, pp. 345–352.

[119] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, pp. 28 – 43, February 2002.

[120] V. Jacobson, "Traceroute."

[121] "Jm," Fraunhofer Heinrich Hertz Institute, Tech. Rep., 2008, [Online; accessed 22-March-2006]. Available: http://iphome.hhi.de/suehring/tml/

[122] X.-J. Zhang, X.-H. Peng, R. Haywood, and T. Porter, "Robust video transmission over lossy network by exploiting H.264/AVC data partitioning," in *Proceedings of the Fifth International Conference on Broadband Communications, Networks, and Systems*, September 2008.

[123] D. Costello and S. Lin, *Error Control Coding*. Prentice Hall, 2004.

[124] A. Ksentini, M. Naimi, and A. Gueroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Communications Magazine*, pp. 107–114, January 2006.

[125] N. Cranley, T. Debnath, and M. Davis, "An experimental investigation of parallel multimedia streams over IEEE 802.11e wlan networks using TXOP," in *Communications, 2007. ICC '07. IEEE International Conference on*, T. Debnath, Ed., 2007, pp. 1740–1746.

[126] N. L. for Applied Network Research / Distributed Applications Support Team, "Iperf," June 2008, [Online; accessed 29-November-2008]. Available: http://dast.nlanr.net/Projects/Iperf/

[127] C. Technologies, "Airpcap wireless capture adaptor," June 2008, [Online; accessed 02-June-2008]. Available: http://www.cacetech.com/products/airpcap_family.htm

[128] S. Glassman, "A caching relay for the world wide web," in *Computer Networks and ISDN Systems*, 1994, pp. 69–76.

[129] M. Sayal, Y. Breitbart, P. Scheuermann, and R. Vingralek, "Selection algorithms for replicated web servers," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 3, pp. 44–50, 1998.

[130] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: algorithms, measurements, and implications," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 41–54.

[131] N. Hu, L. Li, Z. Mao, P. Steenkiste, and J. Wang, "A measurement study of internet bottlenecks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, March 2005, pp. 1689–1700.

[132] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1310–1319 vol.3, Mar 1999.

[133] Z. Miao and A. Ortega, "Scalable proxy caching of video under storage constraints," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 7, pp. 1315–1327, Sep 2002.

[134] J. Kangasharju, F. Hartanto, M. Reisslein, and K. Ross, "Distributing layered encoded video through caches," *IEEE Transactions on Computers*, vol. 51, no. 6, pp. 622–636, Jun 2002.

[135] B. Korte and J. Vygen, *Algorithms and Combinatorics Combinatorial Optimization Theory and Algorithms*, R. L. Graham, B. Korte, L. Lovasz, A. Wigderson, and G. M. Ziegler, Eds. Springer, 2001.

[136] G. B. Dantzig, "Discrete-variable extremum problems," *Operations Research*, vol. 5, no. 2, pp. 266–288, 1957.

[137] J. Wang, "A survey of web caching schemes for the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 5, pp. 36–46, 1999.

[138] Y. Xu and Y. Zhou, "H.264 video communication based refined error concealment schemes," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 1134–1141, 2004.

[139] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 4, pp. 333–344, 2006.

[140] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM (1)*, 1999, pp. 126–134.

Appendix A

Publications

- "Optimal Caching for Partitioned Video", Richard Haywood and Xiao-Hong Peng. Proceedings of the International Symposium on Multimedia
- "Effect of Path Diversity on the loss performance of UDP Packets over the Internet", Richard Haywood and Xiao-Hong Peng. Proceedings of the Fifth International Conference on Broadband Communications, Networks, and Systems
- "Robust Video Transmission over Lossy Network by Exploiting H.264/AVC Data Partitioning", Xing-Jun Zhang, Xiao-Hong Peng, Richard Haywood and Tim Porter. Fifth International Conference on Broadband Communications, Networks, and Systems
- "On Packet Loss Performance under Varying Network Conditions with Path Diversity" Richard Haywood and Xiao-Hong Peng. Proceedings of the International Conference on Advanced Inforcomm Technology.
- "A Hierarchical Unequal Packet Loss Protection Scheme for Robust H.264/AVC Transmission" Xingjun Zhang, Xiao-Hong Peng, Dajun Wu, Tim Porter and Richard Haywood. Proceedings of the Consumer Communications & Networking Conference.
- "Investigation of H.264 Video Streaming over an IEEE 802.11e EDCA Wireless Testbed" Richard Haywood, Saty Mukherjee and Xiao-Hong Peng. Proceedings of the International Conference on Communications 2009.
- "Improving video transmission quality through a combined data partitioning and path diversity strategy", Richard Haywood and Xiao-Hong Peng. Under consideration for IET Electronic Letters

Appendix B
Brite Configuration file

```
BriteConfig

BeginModel
        Name = 5                    #Top Down = 5
        edgeConn = 1                #Random=1
        k = -1                      #Use -1
        BWInter = 1                 #Constant = 1
        BWInterMin = 10.0
        BWInterMax = 1024.0
        BWIntra = 1                 #Constant = 1
        BWIntraMin = 10.0
        BWIntraMax = 1024.0
EndModel

BeginModel
        Name =  3                   #AS Waxman = 3
        N = 75                      #Number of nodes in graph
        HS = 1000                   #Size of main plane
        LS = 100                    #Size of inner planes
        NodePlacement = 1           #Random = 1
        GrowthType = 1              #Incremental = 1
        m = 2                       #Number of neighboring nodes
        alpha = 0.15                #Waxman Parameter
        beta = 0.2                  #Waxman Parameter
        BWDist = 1                  #Constant = 1
        BWMin = 10.0
        BWMax = 1024.0
EndModel


BeginModel
        Name =  9                   #Router Barabasi-Albert2=9
        N = 20                      #Number of nodes in graph
```

```
        HS = 1000                         #Size of main plane
        LS = 100                          #Size of inner planes
        NodePlacement = 1                 #Random = 1
        m = 2                             #Number of neighboring nodes
        BWDist = 1                        #Constant = 1
        BWMin = 10.0
        BWMax = 1024.0
         p = 0.25                         #Probability of adding links
         q = 0.5                          #Probability of rewiring links
EndModel

BeginOutput
        BRITE = 1                         #output in BRITE format
        NS = 1                            #output to NS-2
EndOutput
```