ABSTRACT

LONE, QASIM BILAL. Bandwidth Allocation for Video Streams Subject To An End-to-End Percentile Delay. (Under the direction of Dr. Harry Perros.)

In this thesis we are concerned with how many video streams can be established over a given path of routers so that the 95th percentile of the end-to-end delay of each of these streams is less than a predefined value. For this, we have used four different video traces from Cisco, two from Presence, one from IPTV, and one from a surveillance application. These traces represent realistic video applications currently been carried over the Internet. Using simulation techniques we established the maximum number of video streams that can be admitted for various link capacities subject to a delay percentile constraint.

Bandwidth Allocation for Video Streams Subject To An End-to-End Percentile Delay

by
Qasim Bilal Lone

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Networks

Raleigh, North Carolina

2012

APPROVED BY:

Dr. Khaled Harfoush	Dr. Mihail L. Sichitiu

Dr. Harry Perros
Chair of Advisory Committee

DEDICATION

To my parents.

BIOGRAPHY

Qasim Bilal Lone was born in Karachi, Pakistan. He got his elementary and high school education at Cadet College Kohat, Pakistan. He went on to get his undergraduate degree in Information Technology from National University of Sciences and Technology (NUST), Pakistan in 2006. Following graduation, he worked as a Research Assistant at School of Electrical Engineering and Computer Sciences (SEECS-NUST) and Stanford University, USA for 16 months before coming to NC State for his graduate degree. He will be graduating with a Masters degree in May 2012.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr Harry Perros for all his help. I would also like to thank my committee members Dr Mihail Sichitiu and Dr Khaled Harfoush for their valuable input and suggestions to improve my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

It is expected that video based communications will dominate the way we interact with each other. According to the second annual Visual Networking Index Forecast 2008-2013 [9] global Internet traffic will increase five-fold to 667 exabytes by 2013, and video will represent 91% of all global consumer traffic. Internet based video calling, video instant messaging and webcam traffic was 308 Peta Byte (PB) for the year 2010 and will grow at a compound annual growth rate (CAGR) of 41% by 2015 [9]. Despite the advancement in the field of computer networks, the Internet still suffers from errors, congestion and losses. Real time services in general and IP based video and telephony services in particular have very stringent delay requirements. It has been suggested in G.114 for International Telecommunication Union Standardization Sector (ITU-T) that the tolerable (one way) delay for real time video and audio services should be between 100 ms to 150 ms [21], though delays up to 200 ms can be tolerated [10].

The competition between Internet Service Providers (ISP) continues to grow. To retain their clients and specially corporations, ISPs sign Service Level Agreements (SLA) which bind them to guarantee certain aspects of the services they provide. Bandwidth allocation and Quality of Service (QoS) are major components of those SLAs, which are based on averages and/or percentile statistics of performance measures related to certain aspects of the services. They also advertise audio and video telephony solutions with uninterrupted and high quality Voice

over IP (VOIP) and video over their network. Network providers charge their customers on leased lines and offered services.

Admission control is also an integral part of ISPs and backbone service providers. It ensures that QoS parameters are fulfilled, and necessary policing is provided at edge routers to block any extra traffic which is over the amount of traffic agreed to transport. In order to define policies for admission control and implement SLAs, service providers constantly monitor their network for available and used bandwidth. Carriers plan capacity upgrades in their network based on the rate of growth of bandwidth utilization of their users.

While there had been a lot of work [11] in monitoring of networks with a view to calculate the available and used bandwidth, a thorough study on estimating the required bandwidth of video based services using traces has not been done to the best of our knowledge. The primary objective of this research is to analyze video traces in order to provide estimates on the required bandwidth that needs to be allocated, so that the end-to-end delay is bounded statistically.

1.1 Motivation

An increasing number of Internet users with different demands on services has kept network service providers and researchers interested in providing methods and techniques for improving QoS. Several standards has been proposed and discussed in IETF.

IntServ [22][34] proposed an end-to-end QoS scheme for each flow on a hop-by-hop basis. This approach burdens core routers in the path as it imposes an overhead on the router due to signaling and particularly due to the fact that the connections need to be refreshed continuously. Differentiated Services (Diff Serv) [25][4] proposed a viable alternative which acquits core routers from the overhead of signaling imposed by IntServ. In DiffServ core routers use the DS field of IP packets and implement buffering and scheduling for transmitting out packets. Several classes of traffic can be constructed using the DS bits at network boundaries. Core routers determine how to forwarded packets using those bits.

Alternate schemes such as Measurement Based Admission Control (MBAC) and extensions,

have been proposed in [27] and [29] in order to reduce the complexity within each core router. These schemes use the same concept as IntServ but replace per-flow states with run-time link load estimates performed in each router. The downside of MBAC is that it still requires significant modification of the existing Internet architecture, as load estimation algorithms must be supported at core routers, and there is still a need to be explicitly involved in per flow signaling exchange.

End point Admission Control (EAC) schemes implement per flow distributed admission control on top of Diffserv. EAC schemes use probing packets for the criterion of admission control, these packets are injected in the network from time to time to measure congestion. A detailed review of these schemes is published in [16] and many improvements using the same principle have been proposed recently see [33][14][13][12].

Multiprotocol Label Switching (MPLS) is based on Cisco's tag switching, which was standardized by IETF [24]. It introduces a connection-oriented structure into the otherwise connectionless IP network. In the MPLS architecture a connection has to be first established before the sender can transmit. This is done using a protocol such as RSVP-TE. A connection is setup based on the availabile bandwidth and the IP packets of the connection are transmitted out of the router's port according to their QoS.

All the above mentioned techniques and many other variants, decide on allowing/rejecting a call on the basis of available resources. An issue that still needs to be resolved, however, is how much bandwidth should be allocated to a stream flow so that its end-to-end delay is statistically bounded. That is, the 95th percentile of the end-to-end delay is less than a give target value. Furthermore, given that a certain amount of bandwidth has been allocated in the DiffServ classes AF41, typically used for video, a second issue that needs to be resolved is how many video connections can be carried without violating the end-to-end delay bound. This latter issue is addressed in this thesis.

1.2 Literature Review

In this section we review the relevant literature on bandwidth techniques.

1.2.1 Classification of Bandwidth Estimation Protocols

We first give a brief overview of the various classes of protocol used for bandwidth estimation and congestion avoidance.

Model Based Versus Rated Based Protocols

To provide performance guarantees we can opt for model based protocols which include admission control, policing, reservation and scheduling mechanisms. These protocols are dependent on the underlined architecture of the network and access to routers.

Another approach is to control the rate at which packets are generated and sent over the network. The rate at which packets are generated needs to be less than or equal to capacity of the network. These rates are generally computed by some feed back mechanism from receiver to sender.

TCP Friendly versus Non TCP Friendly Protocols

Multimedia streams do not use the TCP architecture due to its signaling overhead in connection setup and acknowledgement of every packet. Non TCP friendly protocols lead to an unfair resource usage since TCP flows back off after congestion has occurred, whereas non TCP flows continues to choke the network.

Unicast Versus Multicast Protocols

When considering unicast and multicast protocols for bandwidth estimation an important factor is scalability. The complexity of the problem lies in the heterogeneity of the network conditions near the receivers. This problem compounds as the number of the receivers grows. Also, in

case of congestion a decrease in the transmission rate to all the receivers may be unfair, as not all receivers may suffer due to congestion in a specific path.

Single-rate Versus Multi-rate Protocols

Single-rate is used in unicast traffic between a sender and a recipient. Multicast can be single-rate, in whole case when congestion occurs bandwidth for all the flows is reduced, or multi-rate in which case different bandwidth may be assigned to different flows.

Multi-rate congestion control often uses a layered approach as its bases for operation. That is, a sender divides its data in to different layers to be sent to different multicast groups. Every receiver joins the largest possible number of groups permitted by bandwidth availability. The quality of data sent to a receiver is higher when it is part of many groups, than when it is part of fewer groups. This feature is most obvious in multicasting video where the more groups the recipient subscribes to, the more layers the recipient receives and hence quality of the video is better [18].

Sender Based Versus Receiver Based Estimation Protocols

To increase the scalability in multicasting some protocols estimate bandwidth at the receiver end. After estimating the bandwidth, it sends control packet to the sender to increase/decrease the amount of data. To implement this approach a sender must have a video compression capability with different rates. This also warrants the need of some storage at the receiver, to store its previous successful/unsuccessful rates in order to avoid loops and abrupt jumps between states.

1.2.2 Congestion Control and Bandwidth Estimation Protocols

In this section, we describe the basic functionality of some of the popular bandwidth estimation protocols.

Rate Adaption Protocol (RAP)

In RAP, described in [23], bandwidth is estimated using the TCP Additive Increase Multiplicative Decrease (AIMD) mechanism. An acknowledgment is sent for every packet to detect packet losses and calculate RTT. When it senses congestion, the available bandwidth is reduced by half. When there is no congestion the sending rate is increased by one packet at a time. It also uses long term and short term RTT to smooth the sending rate and achieve fine-grained delay-based congestion avoidance.

RAP resembles TCP in timeouts but it waits until three duplicate acknowledgments arrive. Hence, it suffers few timeouts as compared to TCP and it acts aggressively when TCP is suffering timeouts by not backing off. RAP was a basic proof of concept for TCP friendly multimedia content delivery protocols. RAP does not address packet losses immediately as it waits for three duplicated acknowledgments, and therefore it deteriorates the quality of video. It also chokes applications running TCP and TCP friendly protocols.

TCP Friendly Rate Control (TFRC) Scheme

TFC is an active probe based bandwidth estimation mechanism. TFRC is based on complex model equations to estimate bandwidth and control congestion for unicast and multicast traffic. A video encoder adjusts the rate of transmission according to the available bandwidth as estimated by the protocol. The TCP-friendly rate control protocol TFRC is described in [26], and its design is based on minimizing the loss of packets. RTT is measured in TFRC in the standard way by reading the timestamps of packets across the network. TFRC follows a slow start period similar to that of TCP until it reaches its maximum available share of the bandwidth without losses. The slow start is ended with a loss event. In this protocol, the receiver generates a report based on RTT calculated from the one way delay. This report is sent to the transmitter for every packet received. The sender computes its new rate accordingly.

TFRC addresses a number of concerns in their equation based design, such as, responsiveness of persistent congestion, avoidance of oscillations, and unnecessarily abrupt shifts in the sending

rate. The design principles followed by TFRC are as follows:

- Do not aggressively seek out available bandwidth. That is, increase the sending rate slowly in response to a decrease in the loss event rate.
- Do not reduce the sending rate in half in response to a single loss event. However, do reduce the sending rate in half in response to several successive loss events.
- The receiver should report feedback to the sender at least once per round-trip time if it has received any packets in that interval.
- If the sender has not received a feedback after several round-trip times, then the sender should reduce its sending rate, and ultimately stop sending altogether.

Bandwidth estimation is performed at the sender's side. It calculates RTT and retransmit time out value. Receiver sends a feedback packet to the sender which contains a sequence number and the time at which packet was received. The sender smoothes RTT using exponentially weighted moving average. It calculates the retransmit time out value using equation 1.1.

$$t_{RTo} = SRTT + 4 * RTT_{var} \tag{1.1}$$

Where RTT_{var} is the variance of RTT and SRTT is the round trip time estimate. RTT_{var} is multiplied by 4 to provide fairness. This value was proved empirically that it provide good results.

The receiver also calculates the loss event rate p, and sends it to the sender. Estimated loss measures the loss event rate rather than the packet loss rate, where a loss event can consist of several packets lost within a round-trip time. The loss interval is defined as the number of packets between loss events. The estimated loss event rate decreases only in response to a new loss interval that is longer than the previously-calculated average, or a sufficiently-long interval since the last loss event.

TFRC is comes with many improvements to TCP friendly protocols to transmit traffic over the network. However, it suffers the setback due to constant generation of feed back packets. The calculation is fairly complex and does not realize the abruptness required for real time traffic.

The Streaming Media Congestion Control Protocol (SMCC)

SMCC described in [20] relies on bandwidth estimation. It adjust the rate of transmission according to estimated bandwidth. Unlike the above mentioned protocols for bandwidth estimation, SMCC does not go through a slow start phase of TCP, which allows it to send at a relatively better rate which is suitable for streaming media applications. It does not use a congestion window and use continuos probes to monitor congestion and reach to maximum rate. It sends one extra packet per RTT as long as no congestion is detected. When it detects a packet loss it reduces the packet rate to the last Bandwidth Share Estimate (BSE). BSE is calculated using the inter-arrival time between two subsequent packets at the client(receiver). It first calculates the bandwidth sample using equation 1.2

$$Bandwidth = s_2/(a_1 * a_2) \tag{1.2}$$

Where s_2 is the size of the second packet, a_1 and a_2 are the arrival times of the first and second packet respectively. This bandwidth sample is plugged into an exponential filter defined in TCP Westwood [28] to calculate the BSE. It gives us the rate to be used by the sender in order to share bandwidth among flows. The difference between the rate in SMCC and in TCP Westwood is that SMCC uses the inter-arrival time between two successive packets where as TCP Westwood uses the arrival times of acknowledgements for bandwidth estimation. This makes SMCC advantageous in measuring the rate on the forward path while neglecting the effect of congestion on the reverse path. SMCC is also fair to the flows that can be shared; moreover SMCC is friendly to TCP New Reno protocol. SMCC is also robust in responding to packet losses that is due to random errors which is characteristic in wireless connections; this

makes SMCC advantageous due to the growing Internet wireless access.

The Distributed Media Streaming Congestion Control(DMSCC) Scheme

DMSCC described in [17] is designed for estimating bandwidth and congestion control for multiple senders that collaboratively and simultaneously stream media content to a receiver. Each of these senders generates a separate flow to the receiver that is TCP-friendly. Greedy users may increase the number of TCP-friendly flows to acquire a larger share of the bandwidth. DMSCC introduces the task-level congestion control concept. It makes group of flows belonging to the same task to be TCP-friendly, instead of achieving friendliness on each flow individually. DMSCC observes congestion in a distributed media streaming system, identifies the set of flows causing it, and then dynamically tunes these flows to make their over-all throughput TCP-friendly.

DMSCC is a receiver driven protocol, where receivers pull data from multiple senders. It knows the list of senders which are offering the video stream. When a packet loss is observed the receiver computes the correlation function on the delay for various senders. If the correlation is close to equal to 1 it means that a bottleneck is also observed on flows to the selected sender. It selects the sender whose correlation for delay with current sender is close to or equal to 0. One limitation is that all the senders should have same video stream, which makes this protocol unsuitable for real time services. Receiver uses AIMD to increase and decrease the rate at which the data is requested. DMSCC also suffers performance degradation in bursty and lossy environments.

The Dynamic Video Rate Control (DVRC) Scheme

DVRC is a bandwidth estimation protocol which adjusts the rate of video streams using quantization factors to counter congestion and adjust bandwidth for video delivery. Quantization is a lossy step in video encoding which can reduce the size of a frame according to given step size. It reduces the size of video without a big impact on quality of the video. More details on

quantization are given in Chapter 2.

In DVRC each datagram is acknowledged by the receiver. These acknowledgement are not used to enable retransmissions. Rather, they are used to determine the bandwidth and RTT, which are subsequently used to adjust the rate of transmitted video according to the estimated bandwidth.

DVRC is a receiver based protocol which means estimation is performed on the receiver. Prior to sending data a selected number of scale values is assigned to various video encoding methods and transmission policy. The receiver senses congestion, and accordingly it increases or decreases the scale using following scenarios.

- if no congestion is sensed, the scale value is increased by 1
- in case of congestion the scale value is decreased by 1

In order to estimate the bandwidth, the receiver sends control packets to the sender. Each control packet also contains the scale factor. An example of scale factors is given in the table 1.1.

Table 1.1: Experimental Video scale assignment

Scale Value	Average Bitrate (Kbps)
4	340
3	256
2	170
1	86

DVRC also uses layers which are enhancement layers added or dropped depending on network conditions. The video stream is encoded at a base layer and one or more enhancement layers, which can be combined to render the stream at high quality. DVRC supports a varying

number of video scales which are predetermined according to the transmission policies of a specific application.

The Rate control mechanism for packet video in the Internet

This protocol [5] uses a feedback mechanism based on the state of network to control the output rate of video coders. In this approach a receiver indicates with a negative acknowledgement (NACK) packet the loss of a packet, if the number of receivers is less than a threshold. The problem with NACK packets is that in a large multicast network there will be huge amount of negative acknowledgments when congestion starts to appear, which will further aggravate the problem. This is refferd to as the NACK explosion problem. In view of this, large multicast networks (more than 10 receivers), the receiver sends back a feedback message with a quality of service (QoS) measure instead of NACK. This measure is the average packet loss rate they observed during a time interval of length T. The value of T was set empirically as the time required for the receiver to receive 100 packets. The feedback rate has to be at least once ever 2 minutes.

Using the quantization step function the algorithm assigns max rate and calculates the median loss rate. If the median of the loss is above the limit of tolerable loss rate, it decrease max rate by half, otherwise it increases by a fixed fraction of its current value.

1.3 Summary of thesis

Chapter 2 gives an overview of the video technology. The main results of this thesis is in Chapter 3 where we address the issue of how many video connections can be admitted so that a given percentile or end-to-end delay is satisfied. Conclusion and future research are given in Chapter 4.

Chapter 2

An Overview of Video Technology

In this chapter we provide the necessary background information on video compression.

2.1 Introduction

Internet is a viable alternative to distributing real-time multimedia and many new media technologies have been proposed in past few years. Efficient audio and video compression techniques have been proposed in order to meet specific network-oriented constraints, such as, minimize bandwidth, improve multimedia quality and handle other network conditions. Examples of these compression schemes are MPEG-4, H.264/AVC/SVC, G.723, etc. These schemes were designed to adapt to a certain set of network conditions and, in turn, render a minimum quality of service (QoS). In this Chapter we will visit some of the major concepts of compression techniques for video traffic. We will start with basic color theory and then explore some of the main features defined in video transmission in general and H.264 standard. As shown in Figure 2.1 the process of video transmission can be subdivided in to a) sampling of video from camera, b) encoding the video stream in order to compress it at the sender, and c) decoding it at the receiver's end. In the following sections we will explain these steps in more detail.

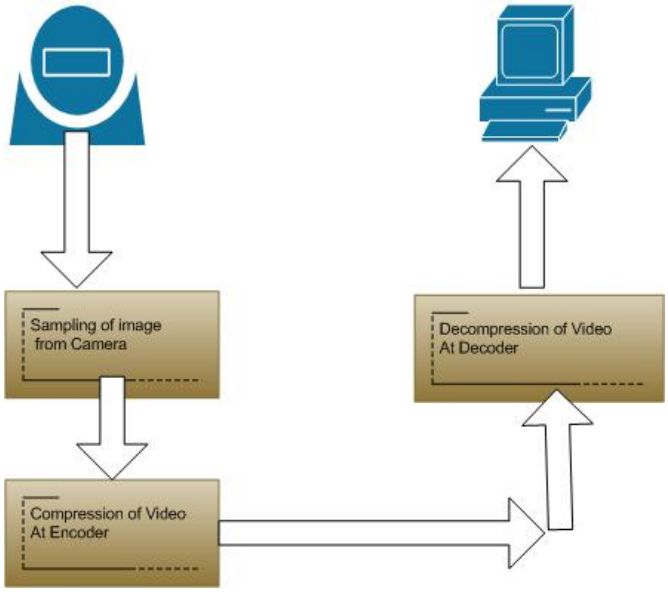


Figure 2.1: An overview of Video Compression[15]

2.2 Sampling of Video

A stream of video traffic is comprised of a series of still frames played one after the other. Human eye regard these still frames as video when they are continuous and played at a speed of 25 to 30 frames per second. Hence in order, to capture this traffic the software takes samples every 1/25 or 1/30 of a second. Sampling takes place in the following domains.

2.2.1 Spatial Sampling

This type of sampling takes place within a frame of a picture.A set of sampling points are defined within the frame by imposing a rectangular or squared grid on it. The sampling points are positioned at the intersection points of the grid. The sampled image is reconstructed by representing each sample as a square picture element called a pixel. The number of sampling points influences the visual quality of the image. If we have more sampling points or smaller sized squares in the grid, then the picture quality will be better; reducing them would make the result in scrambled picture.

2.2.2 Temporal Sampling

Temporal sampling deals with the motion of the picture. Samples are taken at periodic time intervals to produce still frames, which when played in a series produces the motion in video. In this case the trade off is again between size of traffic/bandwidth requirement and smoothness of motion. Higher sampling rate produces smother motion but at the cost of larger data to be transferred over the medium. If the sampling rate is lower, data size will be smaller and hence lower bandwidth requirement to transfer the data, but the motion will be jerky.

2.3 Color Theory for Video Traffic

In the section of spatial sampling, a brief overview was given on how the grid is drawn to take samples of a still frame. Each square of this grid is represented as a pixel, which can be represented by single number if the frame is a monochrome. However, if the frame is a color image we need at least three numbers to estimate the color information of the pixel. Two popular schemes to represent the colors of an image are RGB and YCrCb.

RGB

In this color space a mix of RGB (Red, Green and Blue) can be used to represent any color. When the image is sampled in the spatial domain for RGB method, the color value of each pixel is defined by varying the values of red, green and blue. When the image is displayed the frame illuminates the red, green and blue components of each pixel according to the intensity of each component.

YCrCb

The limitation with the RGB color scheme is that it fails to completely convey the luminance of an image. In the RGB color space, all the three colors are equally important and so are usually all stored at the same resolution. It is possible to represent a color image more efficiently

by separating the luminance from the color information and representing luma with a higher resolution than color, as it is done in the YCrCb color space, a popular way of efficiently representing color images. The luminance component Y is calculated as a weighted average of R, G and B as follows.

$$Y = krR + kgG + kbB, \tag{2.1}$$

where k represents the weight of each of the colors. The color information can be represented as the color difference (chrominance or chroma) components, where each chrominance component is the difference between R, G or B and the luminance Y:

$$Cr = R - Y \tag{2.2}$$

$$Cb = B - Y \tag{2.3}$$

$$Cg = G - Y \tag{2.4}$$

The complete description of a color image is given by Y which is the luminance component and three color differences Cr, Cb and Cg that represent the difference between the color intensity and the mean luminance of each image sample.

The next step is to select a sampling format for the YCrCb scheme. The depiction of the luma value Y, with higher resolution results into more data after compression.

In Figure 2.2 we show three different sampling formats. In the case of 4:2:0 we have 4 values of Y and only two values of chroma values Cr and Cb. In the case of 4:2:2 we sample 2 values of Cr and Cb for every 4 values of Y. In 4:4:4 we save the complete chroma and Y values of the sample.

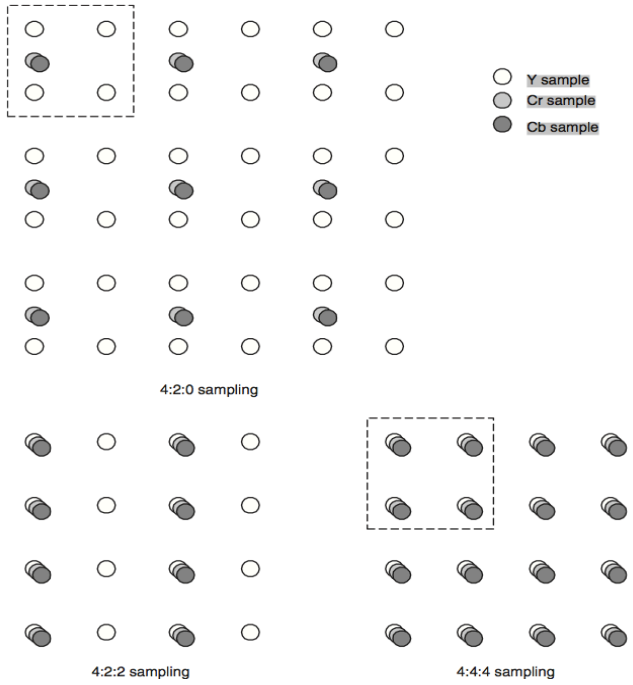


Figure 2.2: Different Formats available for YCrCb Sampling of Image [15]

Let us consider an image with resolution of 720 x 576 pixels and assume that each component (Y,CrCb) is represented by 8 bits. In the case of 4:4:4 we will have 720 x 576 x 8 + 720 x 576 x 8 + 720 x 576 x 8 = 9953280 bits for each Y, Cr and Cb components.

Now the same image sampled with a 4:2:0 scheme, Cr and Cb will result to half the size, that is, (720 x 576 x 8) + (360 x 288 x 8) + (360 x 288 x 8) = 4976640 bits

2.3.1 Intermediate Formats

The Common Intermediate Format (CIF) is the basis for a popular set of formats, which are used to define the spatial sampling of data. The rate of sampling defines the resolution of the video. A list of CIFs along with the resolution are given in the table 2.1. These enteries refer to 4:2:0 format and 8 bits per sample. The choice of frame resolution depends on the application and available storage or transmission capacity. For example, 4CIF is appropriate for standard-definition television and DVD-video; CIF and QCIF are popular for videoconferencing

applications; QCIF or SQCIF are appropriate for mobile multimedia applications [15].

Table 2.1: Video formats, resolution and Bits Per Frame (BPF)

Format	Luminance Resolution	BPF
Sub-QCIF	128 x 96	147456
Quarter-CIF (QCIF)	176 x 144	304128
CIF	352 x 288	1216512
4CIF	704 x 576	4866048

2.3.2 Video Quality Measurement

Quality of a video is a subjective term and is most accurately defined by the user. It also depends upon the application used and circumstances as well. For example in a video conference call a video with a few frames scrambled or not so detailed video might be acceptable as the data transfer is over the Internet. However, for a movie these parameters might be important to the user. Peak Signal to Noise Ratio (PSNR) is one of the test that can be performed to compare the visual deterioration of input of pixel values before they are compressed by the encoder to the output value of the pixel after decoder has decompressed. PSNR is expressed on a logarathmic scale by equation 2.5:

$$PSNR_{dB} = \frac{10log10(2n-1)^2}{MSE}, \tag{2.5}$$

where n is the number of bits per image sample, *(MSE)* is the mean squared error between an original and an impaired image or video frame, relative to $(2n-1)^2$, the square of the highest-possible signal value in the image, used to normalize different intermediate formats so that a fair comparison can be found.

2.4 Video Compression

Video compression is one of the most important components in real time transfer of multimedia content over the Internet. Let us consider an example of raw video data and bandwidth required to transfer it. If we a have video resolution of 512 x 512 pixels with 24 bits per pixel at 30 frames per sec, sustained bandwidth required will be 512 x 512 x 24 x 30 = 188.74 Mbps.

Video compression enables us to condense data by removing the redundancies in the data. The following types of redundancies in the data which can be removed without significant degradation of video:

1. Spatial Redundancy (neighboring pixels carry similar information and are correlated with each other)

2. Temporal Redundancy (subsequent frames have highly correlated pixels at same location, unless there is a complete scene change)

3. Psycho-visual Redundancy (some information is not significant to human eye)

An encoder is present at the source of the video to perform compression and a decoder is at the receiver side to do the decompression.

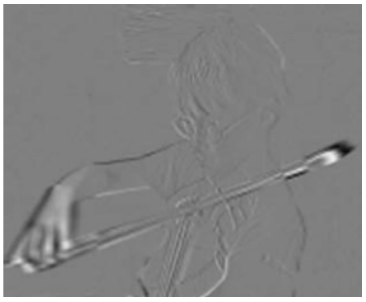
2.4.1 Temporal Redundancy

Temporal redundancy is removed by using the pervious frame as a predictor of the current frame. Subtracting the current frame from the previous frame produces a residual frame. This frame contains values zero, light gray, and dark gray, which depict, positive and negative differences between of the two frames. This residual frame contains less information but still contains a lot of energy, which is present due to movement of the object between the two frames. An example is shown in figure 2.3.

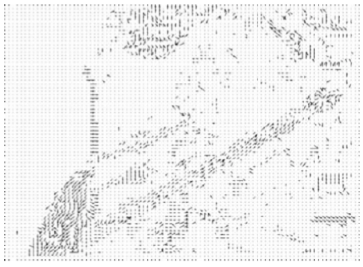
Figure 1(d), shows an optical flow of pixel locations; a vector of movement of objects can further reduce the size of video stream. However, computing this optical flow is computationally

(a): Frame 1 of video

(b): Frame 2 of video



(c): Residual of two frames



(d): Optical flow of pixel locations

Figure 2.3: Residual frame from two consecutive frames [15].

expensive, as we have to calculate the position of each pixel. Also, sending the location of each pixel will create large amount of data which defeats the purpose of compression.

Two methods used to cater for this problem are a) block based motion compensation and b) macro block estimation based on pixel information. In block based motion compensation an M x N block is taken from one frame and is searched in the next. One of the popular searching mechanism is to compare the residual after subtracting the candidate region from the current block and accept or reject based on a given threshold. The block which produces the minimum residual in reference frame is accepted. The residual is then encoded and sent along with motion vector of the entire block to the decoder. The decoder uses this information to predict the location of a block with the motion vector and regenerates the new frame using data from the predictive frame it had received earlier and the residual.

Block based motion compensation is popular because it is computationally straightforward and it can fit rectangular video frames. However, this method is not particularly efficient in

the real world where the objects are not necessarily rectangular or square in nature, do not have predefined edges, and the motion of the objects could be complex. Macroblock motion compensation on the other hand uses pixel matching rather than block matching to locate the movement of the object. In a 4:2:0 format, a 16 x 16 pixel region can be defined by 256 luma, 64 Cr and 64 Cb block. In this method, we estimate the motion of the object by comparing this block with a reference frame to find the closest match. The best-matched region is then subtracted from the reference frame and is sent to decoder along with the vector defining the new position of the block. The decoder uses this information along with data of the previous frame to generate the new frame where the pixel values are shifted according to the vector and re-generated with the help of the residual being added to the same block of previous frame. Macroblock motion compensation can also preform a refined search by performing sub-pixel compensation. This yields a better quality picture but at the expense of more processing and delay in the encoder.

Sub-pixel motion compensation starts the same way as the one implemented by block based search, and after getting the best match, the encoder searches the half-pixel positions immediately next to the best match to see weather the match can be improved. If necessary, the quarter-pixel positions next to the best half-pixel position are also searched. The final block, which could be a pixel, half-pixel or quarter-pixel match along the boundries, is subtracted from the current block or macroblock. Due to extensive matching and incorporating the irregular nature of the objects, the residual produced at the end of the process is significantly less then the one produced by micro-block search or macro-block with integer pixel values.

2.4.2 Spatial Redundancy

As defined earlier spatial redundancy is due to the redundancy with in the frame. An example could be a person standing wearing a black shirt. In this case all pixel values for the black shirt would have approximately the same pixel value and sending the entire region would be waste of resources. To reduce this type of redundancy intra-prediction of frame data as described below

is performed.

2.4.3 Psycho-visual Redundancy

We can eliminate some of the pixel values from the frame and the details, which are lost in the process, are so insignificant that the human eye would not be able to distinguish between the original image and transformed image. Also, if the network has a limited available capacity the encoder is required to send only the significant part of image. In both cases, transform encoding is used to distinguish between the significant and insignificant parts of the image. Quantization (explained latter) is further used to select how much detail needs to be sent.

Figure 2.4 gives an overview of steps required to compress a video. In the next sections we will briefly review each of the required steps.

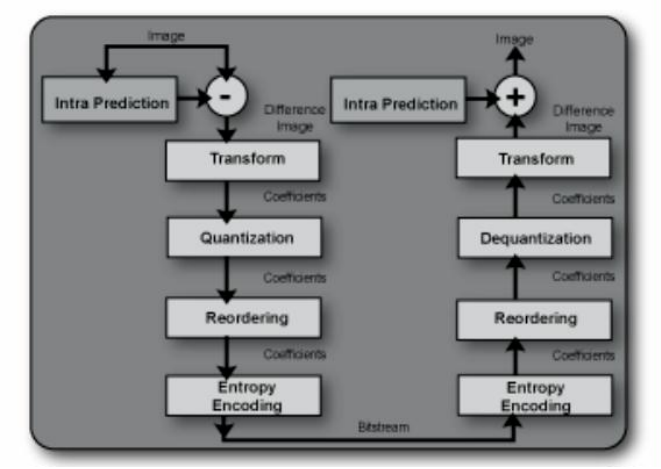


Figure 2.4: Process of Video Compression [31]

2.5 Transform Encoding

In this step we transform our data to a different domain. The selection of the transformation method has to fulfill at least the following conditions:

1. The resultant data generated by the transformation should be de-correlated, i.e. separated into components within minimal inter-dependence, and compact, i.e. most of the energy in the transformed data should be concentrated into a small number of values.

2. At the decoder we would be required to extract the original values so our transformation method should be reversible.

3. The transformation should be computationally achievable, e.g. low memory requirement, achievable using limited-precision arithmetic, low number of arithmetic operations, etc.

4. There are many choices, which fit in the above-mentioned criteria. Some of the well known transformations include Discrete Cosine Transform (DCT), wavelet, Karhunen-Loeve Transform (KLT), etc. We will only explore DCT as it is very popular and it is in MPEG4 and H.264 compression.

2.5.1 The Discrete Cosine Transform (DCT)

DCT works on block by block of the data. It transforms any given block of data by using matrix multiplication. The general form of DCT is given by equation 2.6.

$$Y = AXA^T \tag{2.6}$$

Where X is our input block, Y is our resultant data and A is the transform function.

On the decoder side we can obtain our data back by applying the inverse DCT given by 2.7.

$$X = A^T YA \tag{2.7}$$

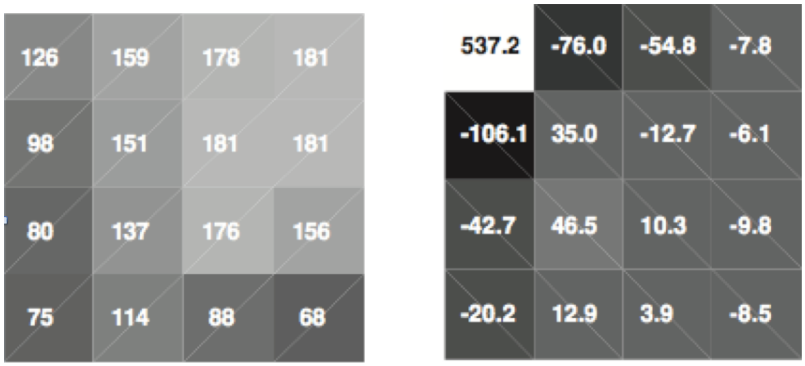
A transform matrix A for 4x4 DCT is a cosine function shown by matrix

$$A = \begin{bmatrix} \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) \\ \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{5\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{7\pi}{8}) \\ \sqrt{\frac{1}{2}}\cos(\frac{2\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{6\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{10\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{14\pi}{8}) \\ \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{9\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{15\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{21\pi}{8}) \end{bmatrix}$$

Evaluating the cosine functions gives rise to the following matrix

$$A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & -0.653 & -0.271 \end{bmatrix}$$

A block by block transformation is applied to the image, and the resultant coefficients are produced. One of the major advantages of applying DCT on the image blocks is that it produces a transformed image with 'energy compaction'. The concept can be explained in more detail with the help of the figures 2.5 and 2.6. In figure 2.5a, we have the original pixel values for a block of image, figure 2.5b shows the DCT coefficients of the same block after transformation.

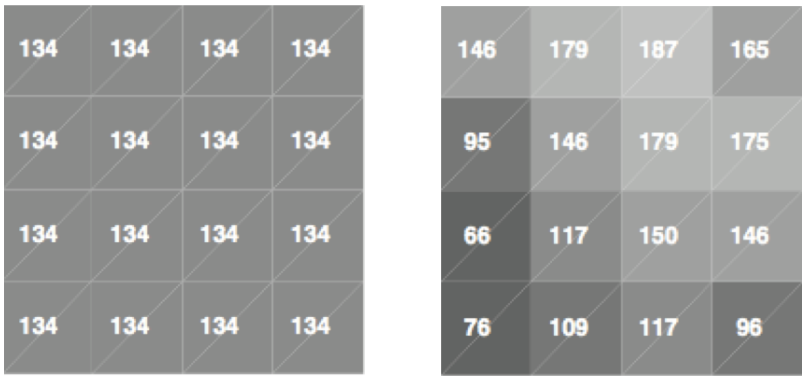


(a): Original Block (b): DCT Coefficients

Figure 2.5: Original Block and DCT coefficients after transformation is applied [15]

Using all the DCT coefficients at the decoder would recover the original block. However,

usefulness of DCT comes if some of these coefficients are dropped. Figure 2.6a shows the values obtained if we use only one most significant coefficient and figure 2.6b shows the values revealed after five coefficients of the most significant values are used.



(a): 1 DCT Coefficient (b): 5 DCT Coefficients

Figure 2.6: Resultant block when 1 and 5 DCT coefficients are used only [15]

It is obvious from the figure that the reproduce values are very close to the original values using just five out of sixteen coefficients values. Quantization uses this property of DCT to reduce the amount of data sent with minimum loss in the image quality.

2.6 Quantization

In this step the data to be transmitted is reduced after DCT has been applied. Encoder controls the flow of transmission by using step size Quantization Parameter (QP). If the step is large, the range of quantized values is small compromising on details of the video quality but achieving high compression. If the step is small, values match the original signal more closely but the larger number of quantized values reduces compression efficiency.

Quantization is a lossy process, which means that data would be lost beyond recovery after the decoder has successfully received a frame and decoded it. The quantizer removes insignificant values such as near-zero DCT or wavelet coefficients and quantizer in an image or

video encoder is designed to map insignificant coefficient values to zero whilst retaining a small number of significant, non-zero coefficients. A technique showed in figure 2.7 below is generally used to locate the parameters, which need to be selected. A zigzag trajectory is followed until we have received all of the parameters or have reached total number of required coefficients.

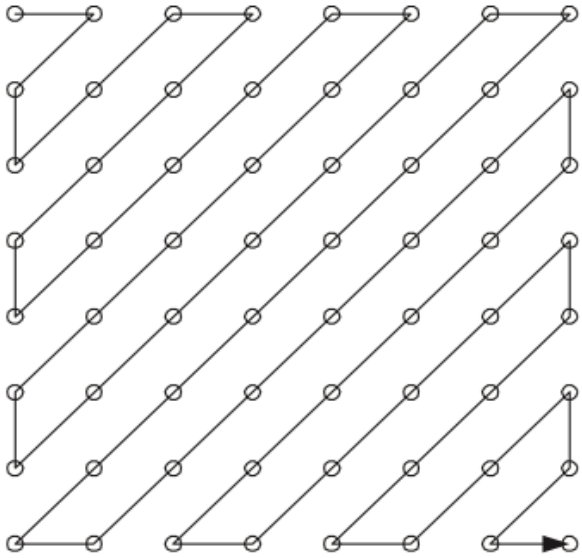


Figure 2.7: Pattern followed for selection of DCT Coefficients [15]

2.7 Entropy Coder

In an entropy coder we use Variable Length Coding (VLC). The idea behind VLC is that there are some values in the data which occur more frequently than others. Therefore, we can save bits if we assign long codes to infrequent values and short codes to frequent values.

There are many coding schemes, which allow us to define and use these codes. The Huffman coding is a popular choice for encoders and decoders. The process works by calculating the probability of occurrence of each symbol and constructing a set of variable length code words in form of lookup table. This table is sent to the decoder so that it can calculate the symbols

correctly. This scheme becomes ineffective for short bursts of video as the overhead could be large or for a very long video because of the unacceptable delay. The encoder needs to calculate the probability table before the decoder can start decoding the data. To overcome this challenge the video coding standard defines code words based on probabilities calculated using generic video material. These pre-calculated tables are already present in an encoder and a decoder and are compiled by the standard body (MPEG 4) by calculating the probability of frequent symbols for a fairly large set of data. The table provides translation between data bits and codes. It follows the same principal to provide smaller codes for frequent data and longer codes for unfrequent data bits.

2.8 Conclusion

In this Chapter we first gave a brief overview of how video compression works, followed by a description of the compression mechanism used by MPEG 4(Part 10) /H.264 standard.

Chapter 3

Simulation and Results

Bandwidth allocation for various types of Internet video is a challenge due to the high variability of the arrival times and bit rates produced by the encoder. As mentioned in Chapter 2, the question of how much bandwidth should be allocated in a model based protocol for congestion control, has not been addressed satisfactorily. In this Chapter we use simulation to investigate how many video streams can be carried over a path of several routers so that a given 95th percentile of the end-to-end delay is satisfied. The simulation is driven by video packet traces obtained from Cisco's telepresence, IPTV, and surveillance cameras.

3.1 Experimental Setup

Our simulation models a network of 5 queues of infinite buffer length as shown in 3.1. Data traffic enters the network from an input stream consisting of packets whose arrival and packet length is based on actual video traces.

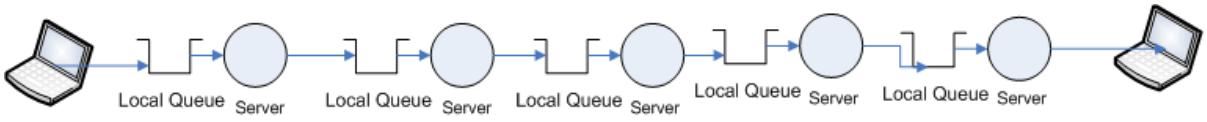


Figure 3.1: The simulated queueing network

The simulator was built in Java programming language. It first parses the video packet trace which is in the form of Packet Details Markup Language (PDML) and then it extracts the arrival time and packet length which is used to generate the arrival process of the packets to the simulator.

The simulator is an event driven model that depicts how packets move through a series of routers. Specifically, we assume that the path between the source and destination consists of five routers. Each router is depicted by a single server queue. The server represents the transmitter of an output port, and the queue in front of it represents video packets waiting to be transmitted out.

The simulation model generates the 95th percentile of the end-to-end delay for a given capacity allocated to the links and a given video packet trace. In addition it generates confidence intervals of the estimated 95th percentile using the batch mean method. In order to obtain good estimates very long simulations are required which exceed the number of packets in the traces. In view of this, when the trace is exhausted it loops back to the beginning of the trace in order to continue the simulation. The length of simulation is determined by the number of packets departed from the queuing network.

For instance let us assume that we run the simulation for 9,000 departures. We divide all departures into batches of 300 arrivals, thus obtaining 30 batches. For the first 300 departures in batch 1, we calculate the 95th percentile of the end-to-end delay time and store it in an array. We repeat this process for the next 300 arrivals without changing anything in our simulation, and so on until you have obtained 30 different percentiles. Now, let T_1, T_2 T_3 ... T_30 be the calculated percentiles from each batch. Then the mean of all percentiles, the standard deviation σ and the confidence intervals are obtained as follows.

$$\bar{T} = \frac{1}{30} \sum_{i=1}^{30} T_i \tag{3.1}$$

$$s = \frac{1}{29}\sum_{i=1}^{30}(d_i - \bar{d})^2 \tag{3.2}$$

$$CI(95\%) = \bar{T} - 1.96\frac{s}{\sqrt{n}}, \bar{T} + 1.96\frac{s}{\sqrt{n}} \tag{3.3}$$

The confidence interval tells us that the true 95th percentile lies within the interval 95% of the time which ensures that if we repeat the above simulation 100 times, 95% of these times, on the average, the true 95th percentile will be within the interval.

We have also analyzed the packet arrival process as it impacts the end-to-end delay. Let a_i denoted the arrival time of the ith packet, and let x_i be the inter arrival time between consecutive packets i and i+1 given by a_{i+1} - a_i. The coefficient of variation squared c^2 for the inter arrival time is calculated from the equation 3.4.

$$c^2 = \frac{s^2}{\bar{X}^2} \tag{3.4}$$

where s^2 and $\bar{X}^2$ are the estimated variance and mean of the inter arrival times.

We also use the traffic intensity in order to gauge the level of congestion in the queues. This is defined as the average number of bits/sec that is generated by a trace divided by a the speed of the link of a router's output port, which is depicted by the service rate at each queue.

Finally, we note that in the simulation model we simulate n, $n \geqslant 1$, identical traces of telepresence, or IPTV, or surveillance. Each trace is looped several times as explained before in order to get reliable statistical estimates. The traces are not started all at the same time. Each new trace starts after a randomly selected number of packets between 30% to 50% of the previous trace has been played out. This randomization of the starting time of each new trace is used to avoid the case where all the traces behave the same at each instance.

3.2 TelePresence

This is Cisco's a solution for virtual face-to-face meetings. At the time of its release, it was the only system which run over existing IP networks in contrast to many others which required a dedicated network [32].

The Cisco TelePresence System provides a number of sub-models for different client needs. Following is the complete set of solutions currently offered [30].

- The Cisco TelePresence System 500 is designed for 1 or 2 users, bringing the Cisco TelePresence virtual in-person experience directly into the private office.
- The Cisco TelePresence System 1000 is for small group meetings and one-on-one conversations.
- The Cisco TelePresence System 1100 is designed for small group meetings and one-on-one conversations in multipurpose conference rooms.
- The Cisco TelePresence System 1300 Series is designed for group meetings in multipurpose conference rooms.
- The Cisco TelePresence System 3010 and 3000 models are for business meetings with up to 6 participants per room.
- The Cisco TelePresence System 3210 and 3200 models are for large group meetings of up to 18 participants per room.

For a single trace the bandwidth requirements change according to the camera resolution and the motion handling features provided by camera. Motion handling is defined as the degree of compression within the encoding algorithm to either enhance or suppress the clarity of the video when motion occurs within the image. High motion handling results in a smooth, clear image even when a lot of motion occurs within the video (people walking around, waving their hands, and so on). Low motion handling results in a noticeable choppy, blurry, grainy, or pixelized

image when people or objects move around [32]. Table 3.1 gives the bandwidth requirement for Cisco Telepresence according to the resolution and motion handling capabilities.

Table 3.1: Bit rate according to resolution and motion handling [32]

Resolution	Frame Rate	Motion Handling	Bit Rate
XGA	5 fps	NA	768 kbps
CIF	30 fps	NA	500 kbps
XGA	30 fps	NA	4 Mbps
720p	30 fps	Good	1 Mbps
720p	30 fps	Better	1.5 Mbps
720p	30 fps	Best	2.25 Mbps
1080p	30 fps	Good	3 Mbps
1080p	30 fps	Better	3.5 Mbps
1080p	30 fps	Best	4 Mbps

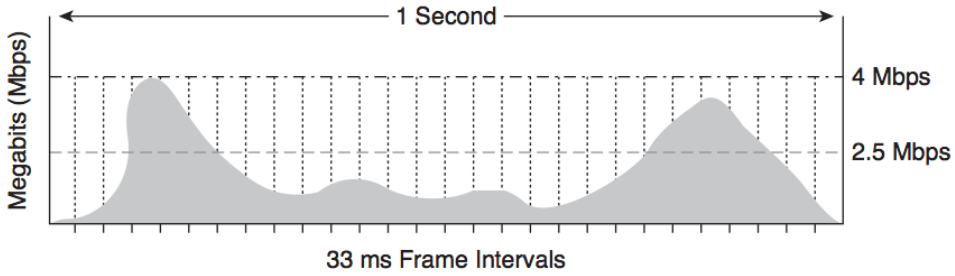


Figure 3.2: Data distribution of audio and video frame [32]

Figure 3.2 shows the bit rate (bps) of a single video with resolution 1090P during 1 second. We observe the variability in the bit rate. High bit rate corresponds to the transmission of I frames. One video frame is transmitted every 33ms. In Telepresence there are only I and P frames. The amount of data transmitted per frame depends on thez change in scene, Figure 3.3 shows a sample video and audio frame sent over the network [32].

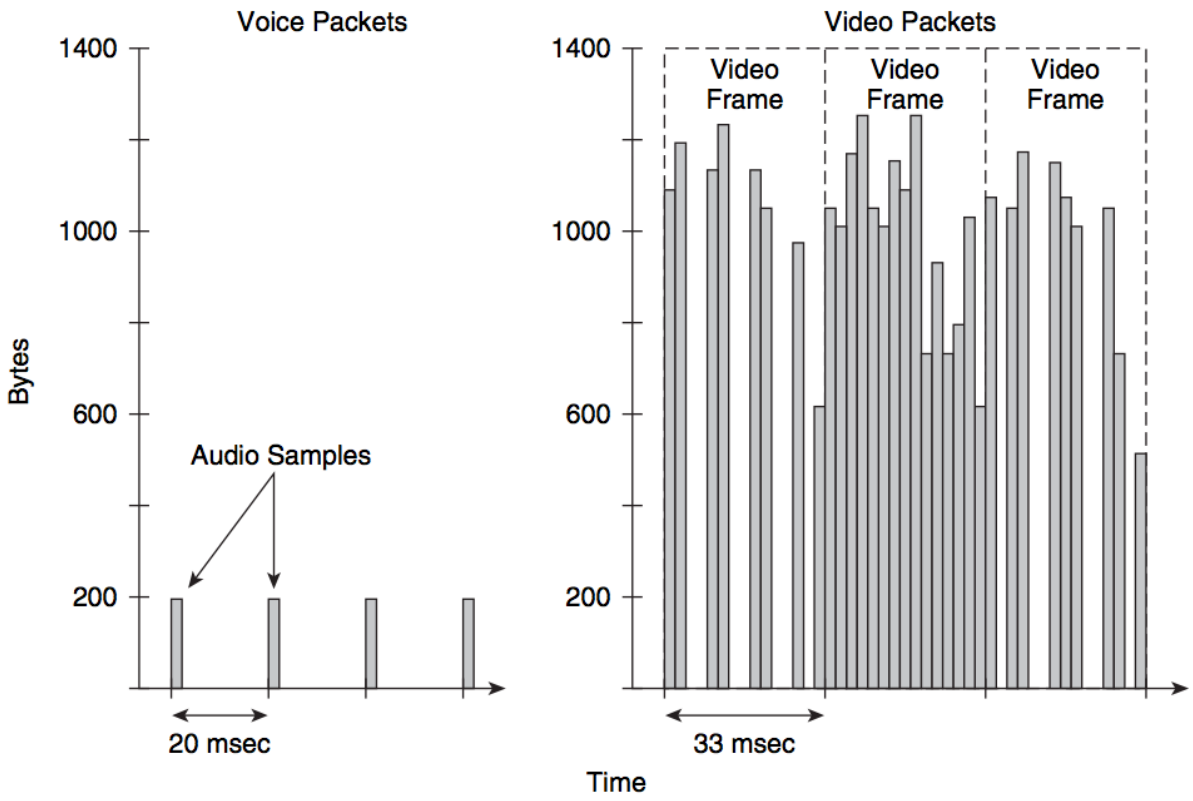


Figure 3.3: Data distribution of audio and video frame [32]

Bandwidth requirements for each of the solutions vary according to the number of streams (video & audio) used by the software. In this section we analyze two different traces taken from telepresence. The difference between the traces is due to the total number of audio and video streams, data per second, inter-arrival time and length of streams. The following traces are used in this analysis.

- Telepresence multipoint-to-multipoint
- Telepresence point-to-point

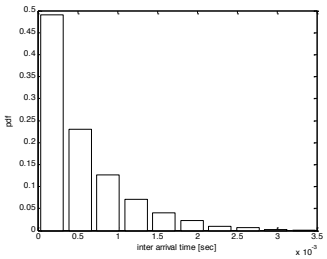
Telepresence multipoint-to-multipoint

In this section we provide some statistics related to the packet arrival process of a single stream extracted from a multipoint-to-multipoint Telepresence trace. Table 3.2 gives a summary of the total number of packets, the average number of packets/sec, the average packet size, the average arrival rate expressed in Mbps, and the squared of coefficient c^2 of the inter arrival time of a single stream. The c^2 is used as the measure of burstiness.

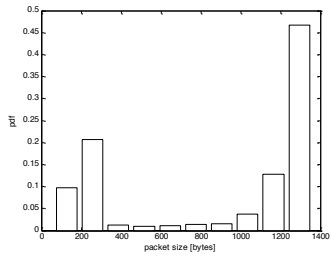
Table 3.2: Summary of the telepresence multipoint-to-multipoint trace

Parameter	Value
No of Packets	177899.0
Avg. No of packets/sec	1270.642
Avg. Packet Size	749.4286 Bytes
Avg. mbps	7.6
C squared of inter arrival time	4.192

A histogram of the inter arrival times is given in Figure 3.4a and in Figure 3.4b the packet length distribution is given.



(a): Histogram of inter arrival times



(b): Histogram of packet lengths

Figure 3.4: Histograms of packet size and inter arrival time

Table 3.3: Admissible streams for various link capabilities

Capacity	Streams								
	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7	Stream 8	Stream 9
Capacity 40 mbps									
ρ	0.2176								
Mean Delay	0.0029								
95th Percentile of Delay	0.0035								
CI on 95th Percentile delay	0.0034,0.0035								
Capacity 80 mbps									
ρ	0.1088	0.1494							
Mean Delay	0.0011	0.0013							
95th Percentile of Delay	0.0014	0.0022							
CI on 95th Percentile delay	0.0014,0.0014	0.0017,0.0028							
Capacity 120 mbps									
ρ	0.0725	0.0997	0.1433	0.1677					
Mean Delay	0.0007	0.0007	0.0008	0.0010					
95th Percentile of Delay	0.0009	0.0009	0.0013	0.0022					
CI on 95th Percentile delay	0.0009,0.0009	0.0009,0.0009	0.0010,0.0015	0.0016,0.0027					
Capacity 160 mbps									
ρ	0.0544	0.0748	0.1075	0.1257	0.1508	0.1674	0.1816		
Mean Delay	0.0005	0.0005	0.0005	0.0006	0.0006	0.0007	0.0015		
95th Percentile of Delay	0.0006	0.0007	0.0007	0.0008	0.0010	0.0015	0.0083		
CI on 95th Percentile delay	0.0006,0.0006	0.0007,0.0007	0.0007,0.0007	0.0008,0.0008	0.0009,0.0011	0.0011,0.0019	0.0057,0.0108		
Capacity 200 mbps									
ρ	0.0435	0.0598	0.0821	0.1050	0.1160	0.1288	0.1505	0.1604	0.1934
Mean Delay	0.0004	0.0004	0.0004	0.0004	0.0005	0.0005	0.0005	0.0006	0.0076
95th Percentile of Delay	0.0005	0.0005	0.0006	0.0006	0.0006	0.0007	0.0008	0.0014	0.0269
CI on 95th Percentile delay	0.0005,0.0005	0.0005,0.0005	0.0006,0.0006	0.0006,0.0006	0.0006,0.0006	0.0006,0.0007	0.0007,0.0010	0.0010,0.0018	0.0186,0.0353

In a multipoint-to-multipoint telepresence, each end point maintains a separate link to all the other end points, thus creating a mesh topology of connections. From the trace we isolated a single one-way video stream which we used in the simulation model of the five node network described earlier in this Chapter. The objective was to estimate how many such single video streams can be admitted so that the 95th percentile of the end-to-end delay of single video streams is less than 50 msec. In the simulation model we did not take into account the one-way propagation delay, since it is fixed. For instance, if the one-way propagation delay is 100 msec, then in essence, we calculate the number of streams that can be admitted subject to 95th percentile of 150 msec. There may belong in the same multipoint-to-multipoint telepresence session, or different multipoint-to-multipoint sessions established over the same path of routers.

The results obtained are summarized in the Table 3.3. For each capacity link we provide the link utilization, the mean end-to-end delay, the 95th percentile of the end-to-end delay, and its confidence interval when n streams are admitted. For instance for a link capacity of 160 Mbps, we see that the 95th percentile of the end-to-end delay is 0.6 msec when 1 stream is admitted, 0.7 msec when 2 streams are admitted and so on. The maximum number of streams that can be admitted is 7 resulting to a 95th percentile of the end-to-end delay of 8.3 msec.

In Figure 3.5, we give the minimum bandwidth required to carry a certain number of connections. We observe that the admissible number of connections grows linearly as we increase the link capacity which is more evident when we fit a linear curve in Figure 3.5. In Figure 3.6 we plot the minimum bandwidth required for Telepresence multipoint-to-multipoint with the bandwidth obtained assuming no statistical gain, that is, the bandwidth for n streams is n times the bandwidth for a single stream. A fair amount of statistical gain is observed as we increase the link's bandwidth.

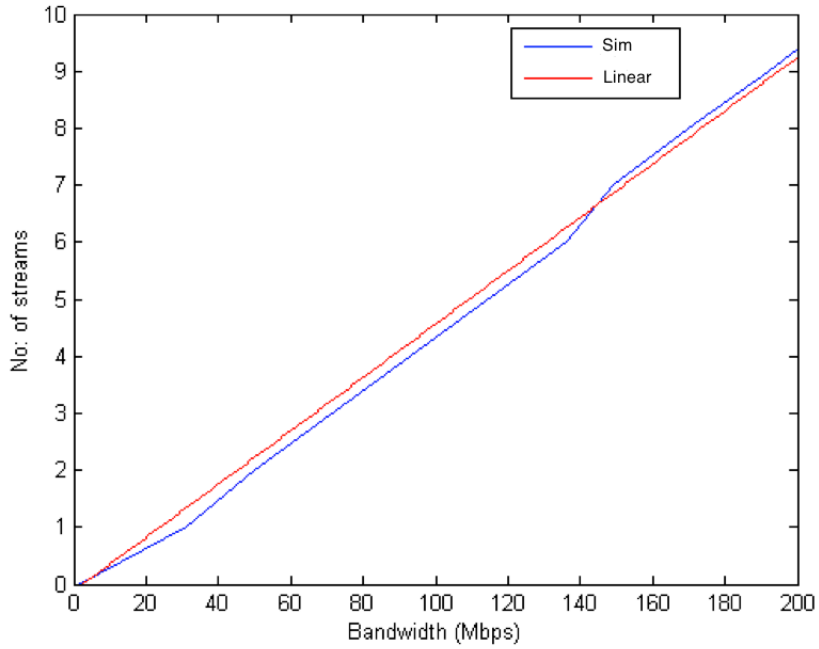


Figure 3.5: Minimum bandwidth required per number of video streams

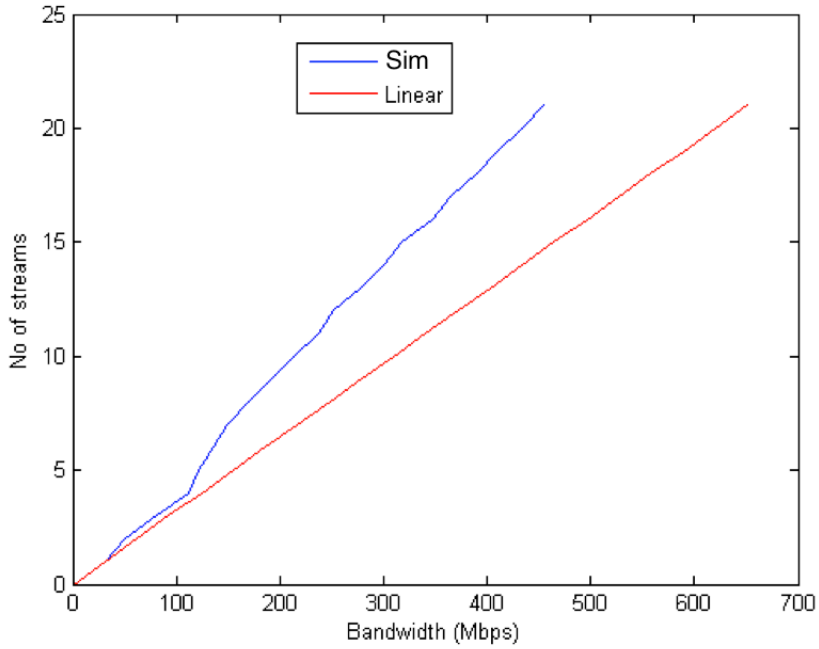


Figure 3.6: Minimum bandwidth required per number compared to linear value for single stream

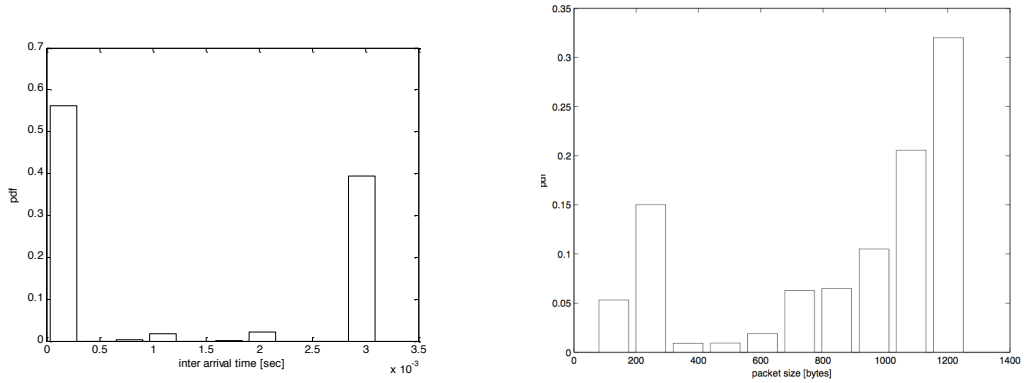
Telepresence point-to-point

A summary of the trace of one-way video transmission is presented in Table 3.4.

Table 3.4: Summary of the telepresence point-to-point trace

Parameter	Value
No of Packets	13259.0
Avg. packets/sec	662.95
Avg. Packet Size	875.72 Bytes
Avg. mbps	4.62
C squared of inter arrival time	1.6597

A histogram of the inter arrival times is given in Figure 3.7a, and the packet length distribution is given in Figure 3.7b. As in the previous case we use the simulation model to compute the maximum number point-to-point video connections that can be admitted over a path of five routers so that the 95th percentile of the end-to-end delay is less than 50 msec. The results are summarized in Table 3.5. Figure 3.8 gives the minimum bandwidth required to support a given number of connections. We also fitted a linear curve in Figure 3.8 and the results are plotted with the data generated by the simulation. We observe that the number of admissible connections grows linearly with the link capacity. In Figure 3.9 we plot the minimum bandwidth reported in Figure 3.8 along with the bandwidh obtained assuming no statistical gain, that is, the bandwidth for n streams is n times the bandwidth for a single stream. Statistical gain is observed as we increase the link's bandwidth.



(a): Histogram of inter arrival times

(b): Histogram of packet lengths

Figure 3.7: Histograms of packet size and inter arrival time

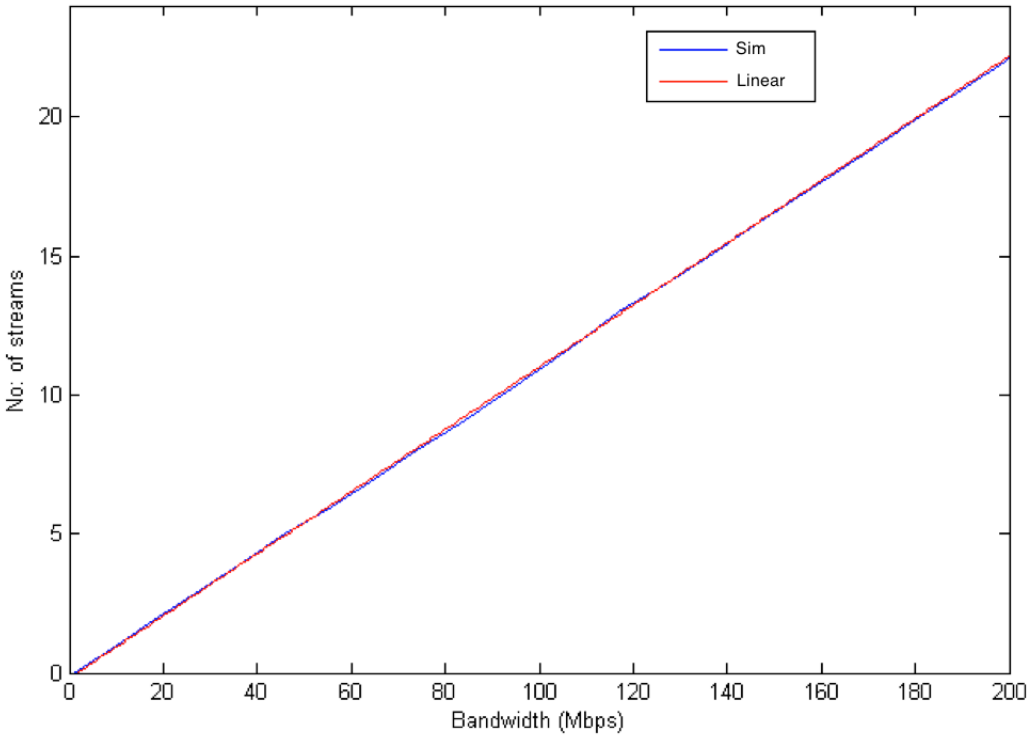


Figure 3.8: Minimum bandwidth required per number of video streams

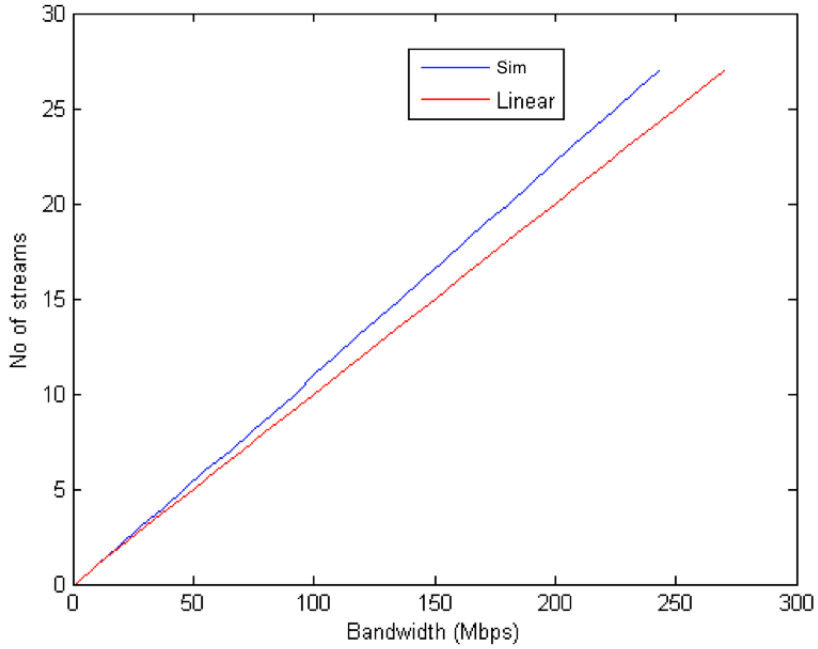


Figure 3.9: Minimum bandwidth required per number compared to linear value for single stream

Table 3.5: Admissible streams for various link capabilities

Capacity	Streams								
	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7	Stream 8	Stream 9
Capacity 10 mbps									
ρ	0.4413								
Mean Delay	0.0158								
95th Percentile of Delay	0.0268								
CI on 95th Percentile delay	0.0262,0.0275								
Capacity 20 mbps									
ρ	0.2206	0.3536							
Mean Delay	0.0048	0.0075							
95th Percentile of Delay	0.0067	0.0143							
CI on 95th Percentile delay	0.0066,0.0067	0.0138,0.0147							
Capacity 30 mbps									
ρ	0.1471	0.2357	0.3275						
Mean Delay	0.0030	0.0033	0.0049						
95th Percentile of Delay	0.0043	0.0048	0.0109						
CI on 95th Percentile delay	0.0042,0.0043	0.0047,0.0049	0.0094,0.0123						
Capacity 40 mbps									
ρ	0.1103	0.1702	0.2367	0.3048					
Mean Delay	0.0023	0.0024	0.0027	0.0039					
95th Percentile of Delay	0.0032	0.0034	0.0044	0.0086					
CI on 95th Percentile delay	0.0031,0.0032	0.0033,0.0034	0.0042,0.0047	0.0074,0.0098					
Capacity 50 mbps									
ρ	0.0883	0.1416	0.1966	0.2281	0.2948				
Mean Delay	0.0018	0.0019	0.0021	0.0023	0.0032				
95th Percentile of Delay	0.0025	0.0027	0.0032	0.0038	0.0073				
CI on 95th Percentile delay	0.0025,0.0025	0.0027,0.0027	0.0031,0.0032	0.0035,0.0042	0.0067,0.0078				
Capacity 60 mbps									
ρ	0.0735	0.1178	0.1638	0.1901	0.2376	0.2762			
Mean Delay	0.0015	0.0016	0.0017	0.0020	0.0020	0.0028			
95th Percentile of Delay	0.0021	0.0023	0.0026	0.0033	0.0033	0.0063			
CI on 95th Percentile delay	0.0021,0.0021	0.0022,0.0023	0.0026,0.0026	0.0031,0.0035	0.0032,0.0035	0.0056,0.0070			
Capacity 70 mbps									
ρ	0.0630	0.1010	0.1404	0.1683	0.1974	0.2298	0.2456		
Mean Delay	0.0013	0.0013	0.0014	0.0015	0.0016	0.0017	0.0023		
95th Percentile of Delay	0.0018	0.0019	0.0021	0.0023	0.0025	0.0030	0.0052		
CI on 95th Percentile delay	0.0018,0.0018	0.0019,0.0019	0.0020,0.0021	0.0022,0.0023	0.0025,0.0026	0.0028,0.0032	0.0046,0.0058		
Capacity 80 mbps									
ρ	0.0552	0.0920	0.1229	0.1578	0.1841	0.2072	0.2344	0.2526	
Mean Delay	0.0011	0.0012	0.0012	0.0013	0.0014	0.0015	0.0016	0.0019	
95th Percentile of Delay	0.0016	0.0017	0.0018	0.0019	0.0023	0.0023	0.0026	0.0041	
CI on 95th Percentile delay	0.0015,0.0016	0.0017,0.0017	0.0018,0.0019	0.0019,0.0020	0.0023,0.0023	0.0023,0.0024	0.0025,0.0027	0.0035,0.0047	
Capacity 90 mbps									
ρ	0.0490	0.0756	0.1053	0.1355	0.1535	0.1842	0.2022	0.2245	0.2455
Mean Delay	0.0010	0.0010	0.0011	0.0011	0.0012	0.0013	0.0013	0.0014	0.0020
95th Percentile of Delay	0.0014	0.0014	0.0016	0.0018	0.0019	0.0020	0.0021	0.0024	0.0043
CI on 95th Percentile delay	0.0014,0.0014	0.0014,0.0015	0.0016,0.0016	0.0017,0.0018	0.0019,0.0020	0.0020,0.0021	0.0020,0.0021	0.0024,0.0025	0.0039,0.0046

3.3 IPTV

The Internet Protocol television (IPTV) is a reliable and secure system through which television services are delivered using the Internet Protocol Suite over a packet-switched network such as the Internet, instead of being delivered through traditional terrestrial, satellite signal, and cable television formats. IPTV services includes Live TV, Video On Demand (VOD) and Interactive TV(iTV) [2].

IPTV has seen a rapid growth in terms of subscribers in the past few years. A report published in June 2010 by Media Research Group, forecasted that the number of global IPTV subscribers will grow from 41.2 million at the end of 2010 to 101.7 million in 2014, a compound annual growth rate of 25.3% [19], see Figure 3.10.

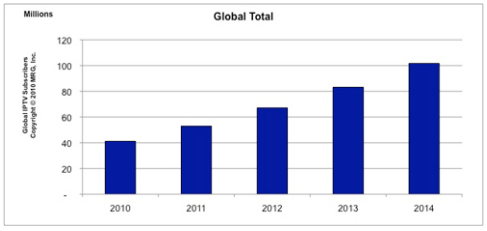


Figure 3.10: Global IPTV Subscriber Forecast [19]

Precept Software were the inventors of IPTV technology. Cisco acquired Precept in 1998 and started as a market leader in this technology [8]. Cisco IPTV software delivers full-screen, full motion video to desktop PCs over private IP networks and the public Internet. The software is multi platform (Unix and Windows) and supports 900-kbps MPEG-1 video stream for Windows desktops; 100-kbps MPEG-4 video stream for low bandwidth Windows users; and 300-kbps RealMedia video stream for UNIX/Linux. It also offers multicast for the subscribers of corporate networks, and unicast for VPN and Internet users.

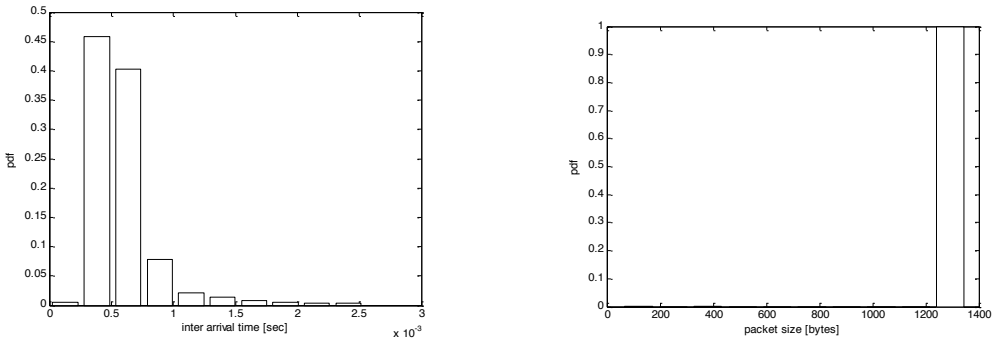
Cisco IPTV has a Content manager, which acts as a policy manager and communicates scheduling information and desired video encoding formats to IPTV broadcast servers. The

broadcast servers encodes and multicast video streams according to directions received by Content manager. The source of stream can be VCR, DVD, satellite and cable feed, or prerecorded media files [7].

Summary of various statistics compiled from the trace are given in Table 3.6.A histograms of the inter arrival time of IPTV given in Figure 3.7a, and a histogram of the packet lengths is given in Figure 3.11b. As can be seen in this case almost all the packets have same length of 1358 bytes.

Table 3.6: Summary of the IPTV trace

Parameter	Value
No of Packets	19349.0
Avg. No. of Packets/sec	1209.3125
Avg. Packet size	1355.191 Bytes
Avg. mbps	13.10
C squared of inter arrival time	1.77



(a): Histogram of inter arrival times (b): Histogram of packet lengths

Figure 3.11: Histograms of packet size and inter arrival time

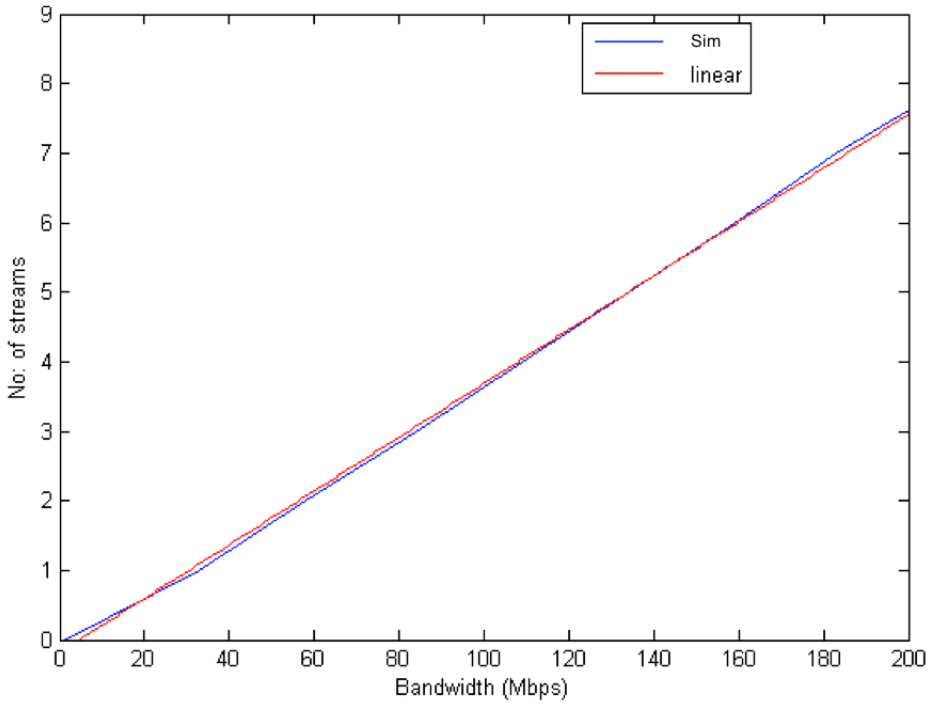


Figure 3.12: Minimum bandwidth required per number of video streams

The results obtained from the simulation are summarized in Table 3.7. Figure 3.12 shows a relationship between the minimum bandwidth required for a given number of connections. We also fitted a linear curve to the simulation results and observed that the data generated by the simulation follows linear curve closely as given in Figure 3.12. In Figure 3.14 we plot the minimum bandwidth required for IPTV with the bandwidth obtained assuming no statistical gain, that is the bandwidth for n streams is n times the bandwidth for a single stream. These values were obtained for mean 95h percentile delay of 50 msecs . Again, as expected, we observe statistical gain.

For slow and lossy network, IPTV offers option of buffering. Typically, the average delay for IPTV is between 0.5 to 2 secs, but it can sustain much larger delays [1]. We have tested number of admissible streams for mean 95th percentile delay of 4 secs. Results obtained from this simulation are given in Figure 3.13. We have also plotted a straight line assuming as before

no statistical gain. We note that when we increase the target 95 percentile to 4 seconds, the bandwidth required for a single stream so that the 95th percentile of the end-to-end delay is satisfied, is close to the average bit arrival of the trace. The results point out that for n streams we require n times the bandwidth of a single stream.

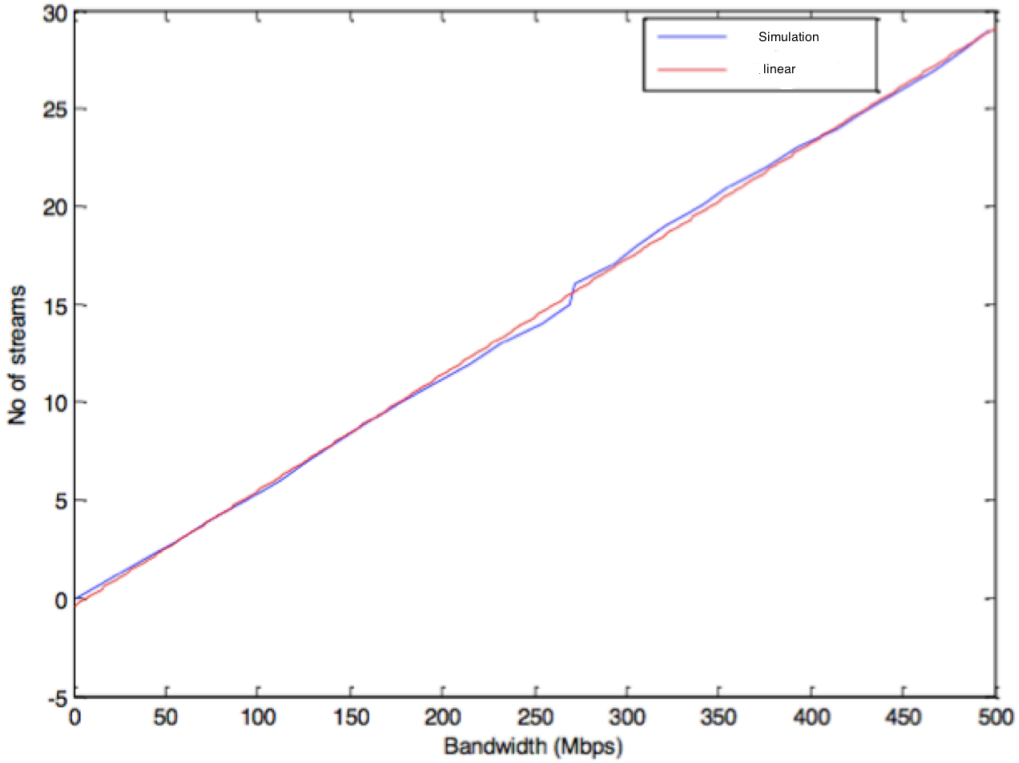


Figure 3.13: Minimum bandwidth required per number of video streams for an avg delay of 4 secs

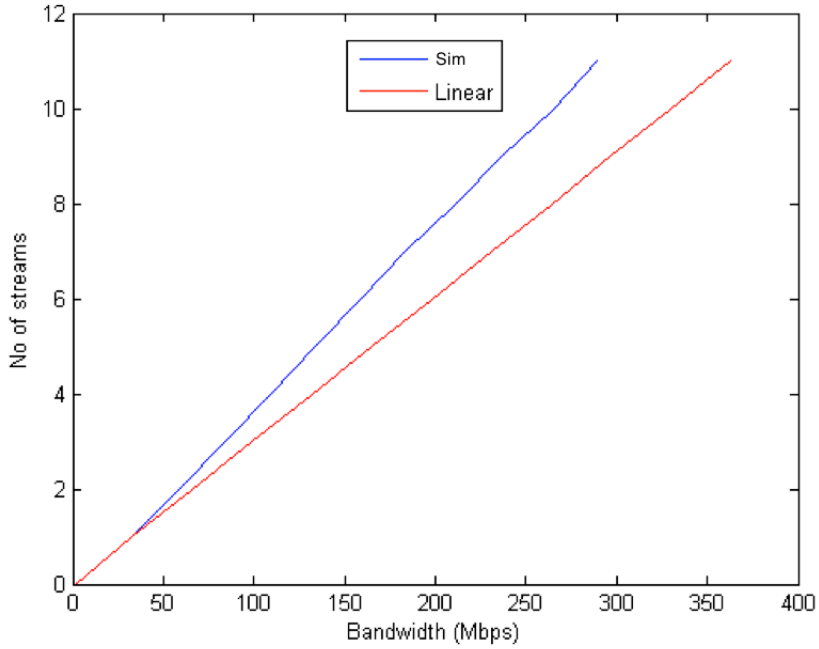


Figure 3.14: Minimum bandwidth required per number compared to linear value for single stream

Table 3.7: Admissible streams for various link capabilities

Capacity	Streams								
	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7	Stream 8	Stream 9
Capacity 50 mbps									
ρ	0.2499								
Mean Delay	0.0021								
95th Percentile of Delay	0.0024								
CI on 95th Percentile delay	0.0024,0.0024								
Capacity 100 mbps									
ρ	0.1250	0.2222	0.2857						
Mean Delay	0.0010	0.0011	0.0019						
95th Percentile of Delay	0.0010	0.0012	0.0057						
CI on 95th Percentile delay	0.0010,0.0010	0.0012,0.0012	0.0049,0.0064						
Capacity 150 mbps									
ρ	0.0833	0.1403	0.1904	0.2319	0.3030				
Mean Delay	0.0007	0.0007	0.0007	0.0008	0.0019				
95th Percentile of Delay	0.0007	0.0008	0.0008	0.0016	0.0082				
CI on 95th Percentile delay	0.0007,0.0007	0.0008,0.0008	0.0008,0.0008	0.0014,0.0017	0.0069,0.0095				
Capacity 200 mbps									
ρ	0.0625	0.1052	0.1500	0.1739	0.2083	0.2400	0.2692		
Mean Delay	0.0005	0.0005	0.0005	0.0005	0.0006	0.0007	0.0019		
95th Percentile of Delay	0.0005	0.0006	0.0006	0.0006	0.0007	0.0016	0.0071		
CI on 95th Percentile delay	0.0005,0.0005	0.0006,0.0006	0.0006,0.0006	0.0006,0.0006	0.0007,0.0008	0.0014,0.0018	0.0064,0.0077		
Capacity 250 mbps									
ρ	0.0500	0.0889	0.1143	0.1454	0.1739	0.2087	0.2240	0.2560	0.2666
Mean Delay	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0005	0.0007	0.0028
95th Percentile of Delay	0.0004	0.0005	0.0005	0.0005	0.0005	0.0005	0.0008	0.0021	0.0119
CI on 95th Percentile delay	0.0004,0.0004	0.0005,0.0005	0.0005,0.0005	0.0005,0.0005	0.0005,0.0005	0.0005,0.0006	0.0007,0.0009	0.0017,0.0026	0.0092,0.0146

3.4 Video Surveillance

Video surveillance is becoming essential in businesses, government organizations and homes. The use of surveillance cameras by governments and businesses has dramatically increased over the last 10 years. In the U.K., for example, there are about 4.2 million surveillance cameras which is about 1 camera for every 14 people [3]. IP based video surveillance is used to transmit captured video over intranet or the internet for storage and analysis purposes.

Cisco offers network-centric video surveillance software and hardware that supports video transmission, monitoring, recording, and management. Cisco video surveillance solutions work with the IP network infrastructure-switches, routers, and other network security devices. Surveillance software is composed of a surveillance manager and a media server. The media server is responsible for sampling of the video from the camera and routing the packets to an IP network. It is also responsible for bandwidth management for distribution of both live and historical recordings. Media server supports standard video codecs such as Motion JPEG, MPEG-4, and H.264 for encoding and decoding of video streams. The software can be integrated with other security and IT applications using open, standards-based APIs. It uses RTP/RTSP streaming as transmission technology to transfer video packets across the network [6].

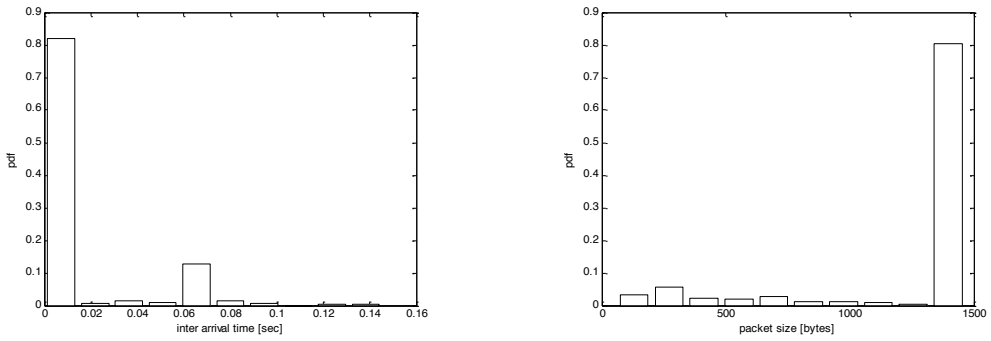
A summary of the statistics obtained from the trace is given in Table 3.8. A histogram of inter arrival times is given in Figure 3.15a, and Figure 3.15b shows the packet length distribution of the Video Surveillance trace. The simulation results obtained are given in Table 3.9. Figure 3.16 indicates the minimum bandwidth required for a given number of connection as a function of the link capacity. We also fitted a linear curve in Figure 3.16 and the results are superimposed on the data generated by the simulation. We observe that the simulation data follows a linear curve. In Figure 3.17 we plot the minimum bandwidth from Figure 3.16 against the bandwidth obtained assuming no statistical gain. We observe a considerable amount of statistical gain.

We also performed simulations when the 95th percentile delay value is 4 secs. Results of the simulations are given in Figure 3.18. As in IPTV, the results appear to follow a linear graph

obtained assuming no statistical gain.

Table 3.8: Summary of the video surveillance trace

Parameter	Value
No of Packets	1439
Avg. packets/sec	75.736
Avg. Packet size	1267.495 Bytes
Avg. mbps	0.765
C squared of inter arrival time	4.6598



(a): Histogram of inter arrival times (b): Histogram of packet lengths

Figure 3.15: Histograms of packet size and inter arrival time

Table 3.9: Admissible streams for various link capabilities

Capacity	Streams									
	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7	Stream 8	Stream 9	Stream 10
Capacity 20 mbps										
ρ	0.0365	0.0633								
Mean Delay	0.0160	0.0168								
95th Percentile of Delay	0.0397	0.0404								
CI on 95th Percentile delay	0.0393,0.0401	0.0398,0.0410								
Capacity 28 mbps										
ρ	0.0261	0.0473	0.0648	0.0763	0.0919					
Mean Delay	0.0101	0.0108	0.0118	0.0146	0.0157					
95th Percentile of Delay	0.0244	0.0250	0.0325	0.0437	0.0475					
CI on 95th Percentile delay	0.0242,0.0247	0.0248,0.0252	0.0297,0.0353	0.0422,0.0453	0.0460,0.0490					
Capacity 32 mbps										
ρ	0.0228	0.0414	0.0543	0.0696	0.0836					
Mean Delay	0.0082	0.0091	0.0093	0.0093	0.0100					
95th Percentile of Delay	0.0197	0.0207	0.0213	0.0236	0.0269					
CI on 95th Percentile delay	0.0194,0.0199	0.0205,0.0208	0.0205,0.0220	0.0214,0.0257	0.0242,0.0295					
Capacity 36 mbps										
ρ	0.0203	0.0351	0.0464	0.0644	0.0716	0.0858	0.0932			
Mean Delay	0.0068	0.0070	0.0080	0.0088	0.0092	0.0111	0.0113			
95th Percentile of Delay	0.0160	0.0163	0.0191	0.0231	0.0276	0.0342	0.0368			
CI on 95th Percentile delay	0.0158,0.0162	0.0159,0.0167	0.0183,0.0199	0.0226,0.0236	0.0261,0.0291	0.0331,0.0353	0.0352,0.0383			
Capacity 40 mbps										
ρ	0.0183	0.0316	0.0435	0.0534	0.0668	0.0745	0.0785	0.0897		
Mean Delay	0.0057	0.0070	0.0072	0.0077	0.0092	0.0094	0.0108	0.0110		
95th Percentile of Delay	0.0130	0.0195	0.0211	0.0224	0.0280	0.0311	0.0344	0.0358		
CI on 95th Percentile delay	0.0128,0.0132	0.0184,0.0205	0.0187,0.0236	0.0212,0.0236	0.0270,0.0291	0.0289,0.0334	0.0336,0.0352	0.0342,0.0375		
Capacity 44 mbps										
ρ	0.0166	0.0301	0.0412	0.0487	0.0585	0.0677	0.0737	0.0904	0.0949	0.1019
Mean Delay	0.0047	0.0049	0.0052	0.0062	0.0082	0.0089	0.0092	0.0092	0.0096	0.0112
95th Percentile of Delay	0.0106	0.0110	0.0125	0.0154	0.0261	0.0296	0.0305	0.0305	0.0318	0.0416
CI on 95th Percentile delay	0.0104,0.0108	0.0107,0.0111	0.0109,0.0143	0.0145,0.0164	0.0252,0.0270	0.0283,0.0309	0.0290,0.0317	0.0293,0.0320	0.0304,0.0332	0.0393,0.0439

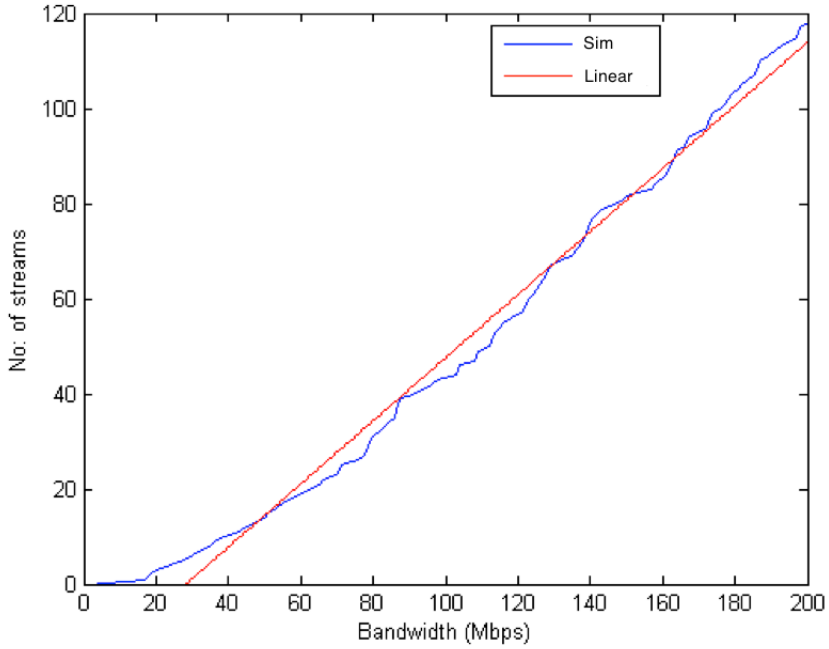


Figure 3.16: Minimum bandwidth required per number of video streams

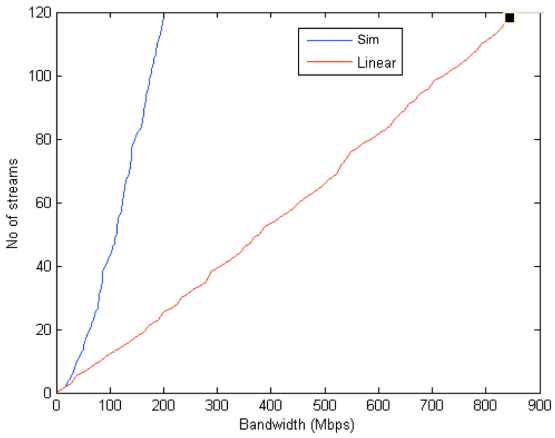


Figure 3.17: Minimum bandwidth required per number compared to linear value for single stream

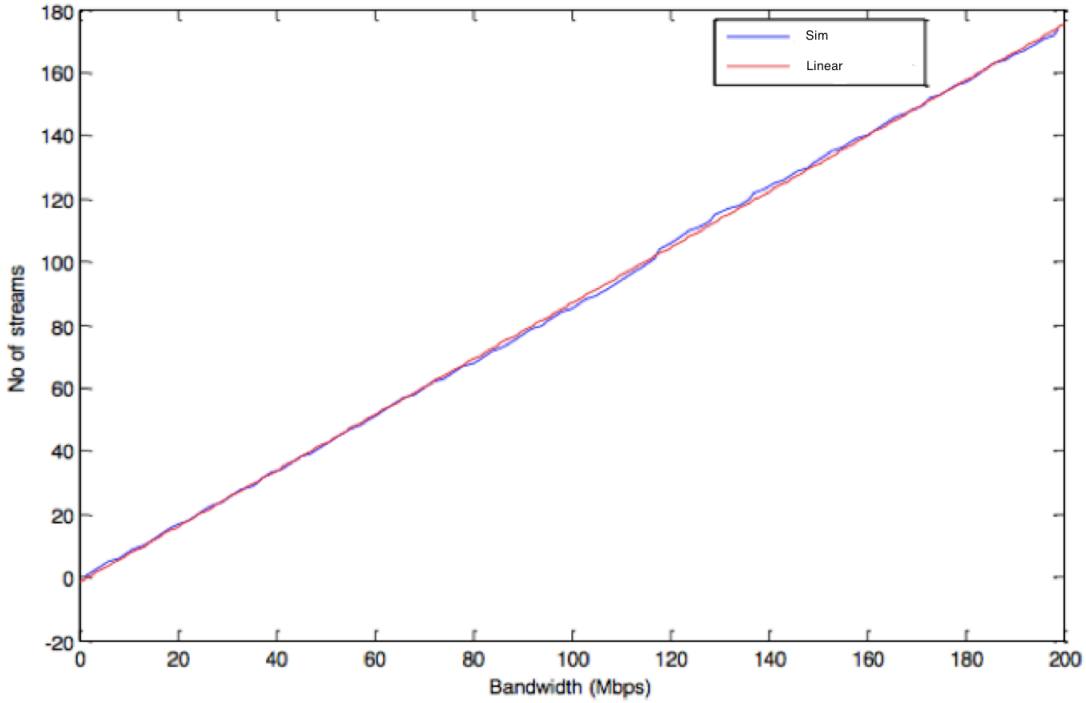


Figure 3.18: Minimum bandwidth required per number of video streams for an average delay of 4 secs

Chapter 4

Conclusions and Future Research

In this thesis we performed simulation and analysis to predict the number of video streams that can be admissible, so that 95th percentile delay is less than a given value. We have found that the admissible number of connections grows almost linearly as we increase the link capacity.

Our results can be used in rate based admission control mechanism to minimize the use of feedbacks, as connections are accepted basis on available resources. Also, in the case of model based approaches, network protocols can make use of initial start phase based on given mechanism, rather than slow start or arbitrary random startup. Our approach can be helpful in admission control at the core routers and supplement smooth delivery of video connections over the network.

This work can be extended in several directions. One potential area is to further investigate results after mixing streams from different video technologies. Also, it would be interesting to see if we can establish a mathematical relationship of how many streams if each type of video can be admitted for a given link capacity so that an end-to-end percentile is satisfied.

REFERENCES

[1] Yingnan Zhu, Wei Liu, Lina Dong, Wenjun Zeng, Yu, H.. High Performance Adaptive Video Services Based on Bitstream Switching for IPTV Systems . *Proceedings of IEEE communications.*, 15:1–5, 2009.

[2] Alliance for Telecommunications Industry Solution. ATIS IPTV Exploratory Group Report and Recommendation to the TOPS Council.

[3] BBC News. Britain is 'surveillance society.

[4] K.Nichols S. Blake F. Baker D. Black. An Architecture for Differentiated Services. *RFC2475*, 1998.

[5] J.-C. Bolot and T. Turletti. A rate control mechanism for packet video in the internet. *Proceedings of the Infocom.*, pages 1216–1223, jun 1994.

[6] Cisco. Cisco Video Surveillance Media Server.

[7] Cisco. How Cisco Designed and Deployed IPTV for Global Communications.

[8] Cisco. Summary of Acquisitions.

[9] Cisco. Cisco Visual Networking Index:Forecast and Methodology 2010-2011. *White Paper*, 2011.

[10] Cisco Systemsi. Cisco medianet Readiness Assessment (MRA) Service. *White Paper.*, 2009.

[11] Cottrell, Les. A comprehensive list of network monitoring tools and links to more information for each tool.

[12] F. Borgonovo, A. Capone, G. Bianchi. Endpoint Admission Control with Delay Variation Measurements for QoS in IP Networks. *ACM SIGCOMM Computer Communications.*, 2002.

[13] F. Kelly, P. Key, S. Zachary. Distributed Admission Control. *Journal of Selected Areas in Communications,*, 2000.

[14] G. Bianchi, A. Capone, C. Petrioli. Packet management techniques for measurement based end-to-end admission control in IP networks. *KICS/IEEE Journal of Communications and Networks (special issue on QoS in Internet)*, 2000.

[15] Donald E. Knuth. *The H.264 Advanced Video Compression Standard.* Addison-Wesley, 2010.

[16] L. Breslau, E. Knightly, S. Shenker, I. Stoica H,. Zhang. Endpoint Admission Control: Architectural Issues and Performance. *Proceedings of ACM SIGCOMM 2000,*, 2000.

[17] L. Ma, W. Tsang. Congestion control in distributed media streaming. *Proceedings of the Infocom.*, pages 1397–1405, 2007.

[18] M. Talat, M. Koutb, H. Sorour. A Survey on Unicast Congestion Control Protocols for Media Traffic. *Proceedings of Internation Journal of Computer Science and Network Security.*, 9(3):254–261, 2000.

[19] MRG Inc. IPTV Global Forecast 2010 to 2014 Semiannual IPTV Global Forecast Report.

[20] N. Aboobaker, D. Chanady, M. Gerla, M. Y. Sanadidi. Streaming media congestion control using bandwidth estimation. *MMNS.*, pages 89–100, 2002.

[21] K. Ramakrishnan. P. Goyal, A.Greenberg. Integration of Call Signaling and Resource Management for IP Telephony. *IEEE Networks*, 13(3):24–32, 1999.

[22] S. Jamin. R. Braden, L. Zhang. Resource Re SerVation Protocol (RSVP)Version 1 Functional Specification. *RFC2205*, 1997.

[23] R. Rejaie, M. Handley, and D. Estrin. Rap: And end-to-end reate-based congestion control mechanism for real-time streams in the internet. *Proceedings of the Infocom.*, pages 1337–1345, 1999.

[24] E. Rosen. Multiprotocol Label Switching Architecture. RFC 3031, January 2001.

[25] W. Weiss. S. Blake, D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and Ipv6 Headers. *RFC2474*, 1998.

[26] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. *Proceedings of the ACM SIGCOMM.*, pages 43–56, 2000.

[27] L. Zhang. S. Jamin P. Danzig S. Shenker. A Measurement-Based Admission Control algorithm for Integrated Services Packet Networks. *IEEE/ACM Transactions on Networking*, 5(1):56–60, 1997.

[28] Savero Mascolo, Calaudio Casetti, M. Gerla, M. Y. Sanadidi . TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. *In Proceeding of ACM SIGMOBILE.*, pages 287–297, 2001.

[29] L. Breslau S. Jamin S. Shenken. Comments on the Performance of Measurement-Based Admission Control Algorithms. *Proc. of IEEE INFOCOM*, 2000.

[30] Cisco Systems. Cisco telepresence system 3010.

[31] Stewart Taylor. H.264 & video compression, Augest 2007.

[32] Tim Szigeti,Kevin McMenamy,Roland Saville,Alan Glowacki. *Cisco TelePresence Fundamentals.* Cisco Press, 2009.

[33] V. Elek, G. Karlsson, R. Ronngren. Admission Control Based on End-to-end Measurements. *Proc. of IEEE INFOCOM 2000*, 2000.

[34] J. Wroclawsky. The use of RSVP with IETF Integrated Services. *RFC2210*, 1997.